View Birdification:

On-Ground Pedestrian Movement Estimation and Prediction from Ego-centric In-Crowd Views

Mai Nishimura

Abstract

Visual Localization is fundamental problem to autonomous navigation which requires reconstruction and maintenance of a map of an unknown environment while at the same time localizing the observer's position and orientation with respect to the map. Localization and Mapping in human-populated environments, in particular, has a wide range of practical applications including automated guidance at the airport and delivery systems in the city. Localization in a crowded scenario has long been a challenge as most conventional studies are built on top of a static world assumption, *i.e.,* we can constantly observe static reference points and solve a geometric reconstruction problem using them. In a highly dynamic environment such as crowded streets, such static backgrounds are often occluded or hard to track due to severe occlusions by pedestrians in the foreground. This brings us to a key research question — *Can we achieve visual localization by using only dynamic objects?*

To answer this question, we introduce *view birdification*, the problem of recovering ground-plane movements of people in a crowd from an ego-centric video captured by an observer (*e.g.,* a person or a vehicle) also moving in the crowd. Contrary to conventional approaches built on the top of the static world assumption, view birdification depends only on the perceived movements of dynamic objects. In this dissertation, we first formulate view birdification as a geometric trajectory reconstruction problem and derive a cascaded optimization method from a Bayesian perspective. The key difficulty underlying this problem is that the two kinds of trajectories, the camera ego-motion and pedestrian trajectories on the ground plane, are deeply intertwined in the observed movements in an ego-centric view. This Bayesian formulation alternately updates the estimated camera ego-motion and pedestrian locations relative to it. We empirically analyze the properties of the solution with regard to the number of pedestrians. Second, we derive a data-driven solver for view birdification with simultaneously learning of an underlying motion model. We refer this method as *ViewBirdiformer*.

ViewBirdiformer is based on a Transformer, which models view birdification as a set-to-set translation problem between the ego-centric and the on-ground views while simultaneously learning the interaction model between pedestrians. Extensive evaluations demonstrate the accuracy of our method and set the ground for further studies of view birdification as an important but challenging visual understanding problem.

Lastly, we extend view birdification as an object-oriented world model. View birdification is the problem of geometric reconstruction of the on-ground pedestrian trajectories while estimating the camera ego-motion. We extend this as a transition model that predicts the future state of the crowd from in-crowd views. Unlike conventional world models that predict the future state of the whole image in an ego view, we aim at constructing an object-oriented world model that can estimate the future states of each pedestrian while learning the interactions between them. We refer to this object-oriented world model as the Pedestrian World Model, a computational transition model of pedestrians that can continuously localize and predict the movements of all people visible to the observer on the same ground plane. To represent this Pedestrian World Model, we derive InCrowdFormer, a Transformer-based architecture that uses attention for pedestrian interaction modeling and egocentric to top-down view transformation and autoregressively predicts on-ground positions of a variable number of people. InCrowdFormer is formulated as a generative model and can encode the uncertainties arising from unknown pedestrian heights with latent codes to predict the posterior distributions of pedestrian positions.

In this dissertation, we introduce problem, theory and methodology of view birdification for on-ground localization and prediction. To tackle this challenging problem in a highly dynamic environment, we construct an evaluation platform consisting of a simulator and real trajectories datasets and validate the effectiveness of our proposed method with a diverse set of crowds. We believe view birdification becomes essential for mobile robot navigation and localization in real-world crowds. Recovered ground-plane movements would provide a sound basis for situational understanding and benefit downstream applications in computer vision and robotics.

List of Publications

Publications included in this dissertation

This dissertation consists of the following four publications.

Referred Conference Proceedings

1. Mai Nishimura, Shohei Nobuhara, Ko Nishino, "View Birdification in the Crowd: Ground-Plane Localization from Perceived Movements", In Proc. The British Machine Vision Conference (BMVC), pp. 256-270, 2021.

2. Mai Nishimura, Shohei Nobuhara, Ko Nishino, "InCrowdFormer: On-Ground Pedestrian World Model From Egocentric Views", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, under review

Referred Journal Articles

1. Mai Nishimura, Shohei Nobuhara, Ko Nishino, "ViewBirdiformer: Learning to recover ground-plane crowd trajectories and ego-motion from a single ego-centric view", IEEE Robotics and Automation Letters (RA-L), pp. 368-375, 2023, to be presented at the International Conference on Robotics and Automation (ICRA), 2023.

2. Mai Nishimura, Shohei Nobuhara, Ko Nishino, "View Birdification in the Crowd: Ground-Plane Localization from Perceived Movements", International Journal of Computer Vision (IJCV), under review (major revision)

Acknowledgement

First, I wish to express my deepest gratitude to my supervisor, Professor Ko Nishino for his continuous guidance as well as insightful discussion. Without his expertise, patience, and constructive criticism, this work would not have been possible. I have learned a great deal from him throughout my research projects and have always been impressed by his work as a leading researcher and also as a great educationist. It was a great fortune for me to have received his guidance, which has made this doctoral program a truly exceptional experience.

Second, I am also grateful to Associate Professor Shohei Nobuhara for his considerate support, not limited to research topics. He had also been my supervisor during my bachelor's and master's courses, and I learned the foundation of computational geometry from him. His enthusiastic guidance enabled me to pursue computer vision until now.

I wish to thank my dissertation committee members, Professor Tatsuya Kawahara, and Professor Takayuki Kanda, for their insightful discussion and valuable feedback, which significantly improved this dissertation.

Completing this Ph.D. program while working full-time would not have been possible without the understanding and cooperation of my company. I am deeply grateful to Dr. Masaki Suwa and Dr. Yoshihisa Ijiri for providing me with this opportunity. I would also like to express my sincere gratitude to Dr. Ryo Yonetani, who served as the Principal Investigator of my research group at the company. Throughout numerous collaborative projects, I learned how to plan, polish, and publish research papers.

I am also grateful to the laboratory staff, Ms. Asako Yoshimura, for her administrative support and kindness. I would also like to extend my gratitude to my friends, Dr. Yoshiaki Bando and Dr. Yoichi Chikahara, for their advice and encouragement, which helped me complete this dissertation.

Finally, I extend my deepest appreciation to my husband, Yuhei, for his understanding and continuous support throughout my journey to this point.

Contents

1 Introduction **13**

1.1 Spatial Perception in a Crowd 13
1.2 View Birdification 15
1.3 Pedestrian World Model 16
1.4 Transformers for Geometric Translation and Object Interaction Modeling 18
1.5 Contributions 19
1.6 Dissertation Overview 21

2 Related Work **23**

2.1 Bird's Eye View Transformation 23
2.2 Dynamic SLAM 24
2.3 Crowd Modeling 25
2.4 Non-rigid Structure from Motion 25
2.5 3D Multi-Object Tracking (MOT) 26
2.6 World Models 26
2.7 Crowd Navigation 27

3 View Birdification **29**

3.1 Background 29
3.2 Geometric View Birdification 31
3.2.1 Problem Setting 31
3.2.2 Geometric Observation Model 32

4 View Birdification from a Bayesian Perspective **35**

4.1 A Cascaded Optimization for View Birdification 36
4.1.1 A Bayesian Formulation 36
4.1.2 Energy Minimization 37

4.1.3 Pedestrian Interaction Models . . . 39
4.1.4 Optimization over a Large Number of Pedestrians . . . 40
4.1.5 Implementation Details . . . 41
4.2 Experiments . . . 41
4.2.1 View Birdification Datasets . . . 42
4.2.2 Example Trajectories . . . 44
4.2.3 View Birdification Results . . . 44
4.2.4 Unknown Ego-Motion Recovery with the Real Mobile Platform Dataset . . . 48
4.3 Discussion . . . 50
4.4 Failure Cases . . . 51

5 Learning to recover ground-plane crowd trajectories and ego-motion 57
5.1 Background . . . 57
5.2 Preliminary . . . 59
5.3 ViewBirdiformer . . . 60
5.3.1 Geometric 2D-to-2D Transformer . . . 61
5.3.2 Relative Position Transformation . . . 62
5.3.3 Ego-motion Estimation by Task-specific Heads . . . 63
5.4 Experiments . . . 65
5.4.1 View Birdification Datasets . . . 65
5.4.2 Comparison with Geometric Baseline . . . 68
5.4.3 Ablation Studies . . . 72
5.4.4 Limitations and Degenerate Scenario . . . 73
5.5 Implementation Details . . . 74
5.6 Handling Occluded Pedestrians . . . 75
5.7 Comparison of Crowd Motion Models . . . 75
5.8 Attention Visualization . . . 76

6 Pedestrian World Model 79
6.1 Background . . . 79
6.2 Pedestrian World Model . . . 81
6.3 InCrowdFormer . . . 83
6.3.1 Vision Module . . . 83
6.3.2 Geometric Memory Module . . . 84
6.4 Probabilistic InCrowdFormer . . . 85
6.5 Experiments . . . 88

6.5.1 Quantitative evaluation of prediction accuracy 92
6.5.2 Inference on Real Data 93

7 Conclusion **95**
7.1 Summary of Contributions 95
7.2 Towards Autonomous Navigation in the Real World 97
7.3 Future Directions 98

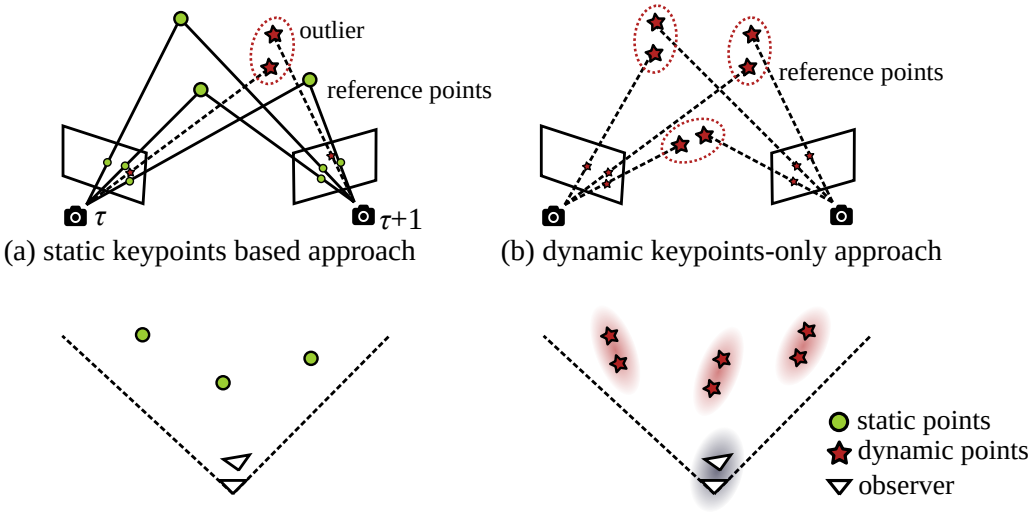
Chapter 1

Introduction

We as human beings are capable of mentally visualizing our surroundings in a top-down view through observation with our eyes. When walking down a crowded street while avoiding potential collision with nearby pedestrians, our mental model lets us localize ourselves on the map, constantly update relative locations to nearby pedestrians, and even predict how they will move. Take another example from playing sports in a team. Professional soccer players are able to make a killer pass as if they perfectly knew the future locations of teammates, while they are also running on the ground. The ability of reconstructing, updating, and predicting object locations in a mental map can be helpful for various applications in our real world full of dynamic objects. Let us refer to this capability of geometric reasoning and prediction working in our subconscious mind as *spatial perception*. We assume that spatial perception is built on top of the intersection of geometric reasoning and prior learnt from our past experiences, *i.e.,* walking in a crowd or playing sports in a team. Throughout this dissertation, we explore this spatial perception as a computer vision task and derive the foundation for modeling a crowded environment from an ego-centric view in the crowd itself.

1.1 Spatial Perception in a Crowd

The spatial perception consists of geometric reasoning abilities to understand the surrounding environment through perception with our eyes. Such a geometric perception of the surroundings allows us to successfully traverse in unknown environments while implicitly constructing the map representation centered around us. This dissertation does not intend to fully emulate the spatial perception of us, which is impossible due to the infinite number of situations in the real world.



(a) static keypoints based approach (b) dynamic keypoints-only approach

(c) localization from static keypoints (d) localization from dynamic keypoints

Figure 1.1: The concept of dynamic keypoints-based reconstruction. (a) Conventional static keypoints-based approach usually filters out dynamic keypoints as an outlier and (c) localizes an observation camera relative to the static reference points. (b) Our dynamic keypoints-based localization fundamentally differs from (a) as it requires only dynamic reference points and (d) recovers trajectories of both the dynamic reference points and an observer.

Specifically, we focus on establishing a foundation for spatial perception in a crowd, where we assume that an observer is immersed in a crowd consisting of people heading towards their destinations while implicitly interacting with each other. Especially in densely crowded environments, we no longer construct a geometrical map of the static background, but instead grasp the relative layout of the surrounding pedestrians as an abstracted map representation. For simplicity, we assume that an observer, *i.e.*, a pedestrian or a wheeled robot, traverses on the planar ground and the observation camera is equipped in parallel with the ground plane. A typical scenario for the spatial perception in a crowd is when a person with a body-worn camera is immersed in a crowd consisting of people heading towards their destinations while implicitly interacting with each other. Our goal is to deduce the mapping from an ego-centric view observation into the geometric perception of the surrounding pedestrians on the 2D ground-plane coodinates while we also walk along the crowd flow.

1.2 View Birdification

Spatial perception involves two major components — Localization and Mapping. While Mapping is to construct a globally consistent map representation of the environment from a local perspective, Localization is to estimate the location and orientation of an observer with respect to the map. These problems have long been extensively studied in the fields of computer vision and robotics, and are the foundations for various downstream applications such as mobile robots. A classical and well-established approach is first to detect and track keypoints in the scene and then reconstruct 3D locations of the keypoints and the observation camera by solving geometric constraints between them[1]. This approach, however, cannot directly be applied to crowded environments full of dynamic objects due to the "static world assumption" [34]. They assume that detected keypoints are static across the frames and can be observed in multiple viewpoints without occlusion. Such an assuption is easily violated in a crowded scene where static backgrounds are often occluded by pedestrians in the foreground. Typical approaches for handling such dynamic scenarios are to eliminate as outliers dynamic points which do not satisfy the assumed geometric constraints [65, 21], or explicitly detect and filter objects by using off-the-shelf object trackers [97, 86]. These approaches are simple yet effective extensions of static keypoints-based approaches for handling a small number of dynamic obstacles. Another line of work clusters static and dynamic objects and associates the dynamic objects with respect to the static map [74], or constructs a factor graph that incorporates both [37, 95]. All of these approaches, however, require a sufficient number of static keypoints that satisfies geometric constraints and do not work for a crowded world where dynamic keypoints are dominant. How can we formulate localization and mapping in such crowded scenes where we can't observe any static keypoints? In other words, can we formulate localization and mapping for highly crowded environments only from the observation of dynamic objects?

To address this question, we formulate *View Birdification*, the task of simultaneously recovering the location of a camera and its surrounding pedestrians only from perceived movements in the video. As Fig. 1.1 depicts, unlike conventional approaches that heavily depend on the static world assumption [34], view birdification aims at modeling a dynamic world by using only dynamic objects as

[1] In computer vision, geometric constraints refer to properties which observed keypoints satisfy by simple geometry, such as linear projection and planar or orthogonal constraints for a group of points.

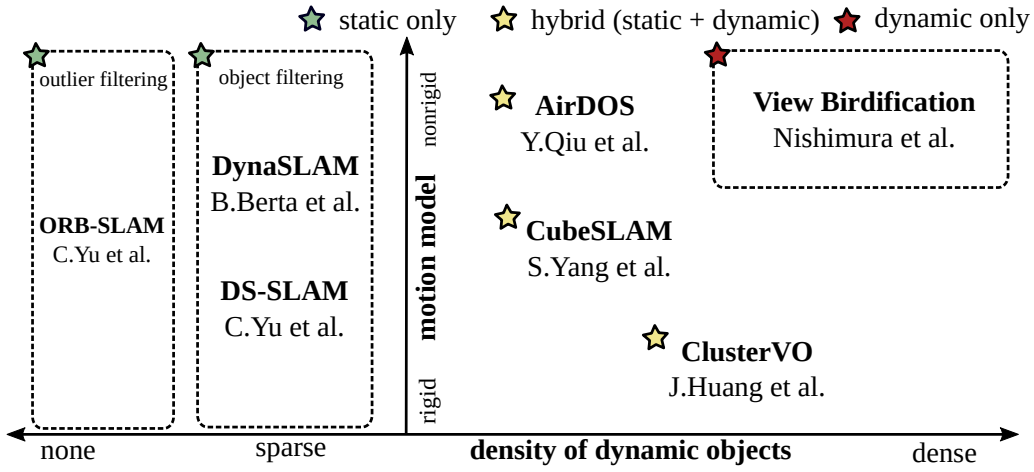
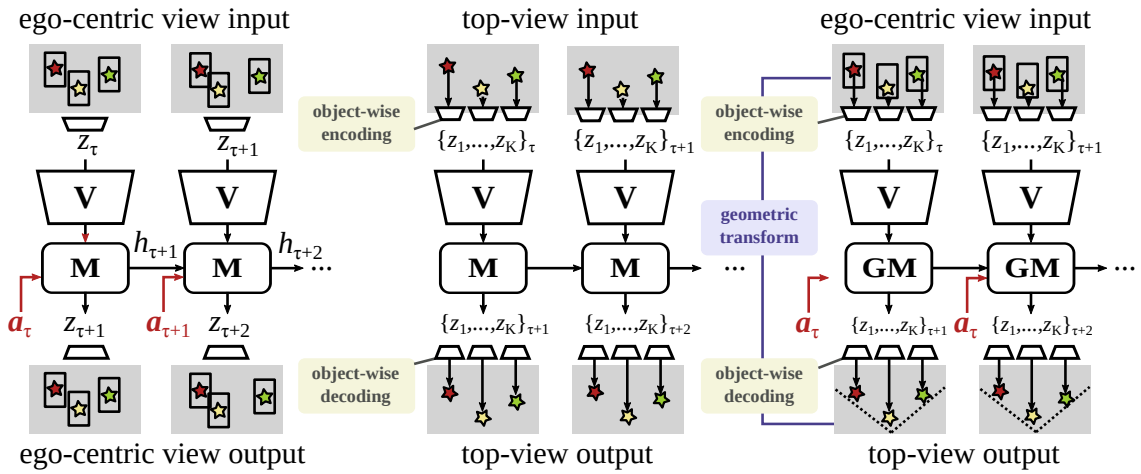


Figure 1.2: Our target environment of interest. Unlike conventional approaches that mainly or partly rely on the static world assumption, view birdification depends only on the dynamic reference objects. This allows us to reconstruct highly crowded environments independent of observation of static background, which is usually hard due to frequent occlusions.

visual cues. The key idea is *objects as dynamic keypoints*. The dynamic world modeled by view birdification is purely object-centric, where mapping reconstructs the global layout of dynamic objects from locally perceived movements. While localization using only dynamic objects is an ill-posed problem in principle, we explore making this well-posed problem with several practical assumptions. One key assumption is that pedestrians follow a common motion model *i.e.,* dominant crowd flow in highly crowded environments, which provides a powerful constraints to resolve the complex localization problem. Fig. 1.2 illustrates our target environment of interest. Our work is the first to explore localization in a crowd without the static world assumption just from in-crowd perception.

1.3 Pedestrian World Model

What we perceive is governed by the prediction of the future based on our mental model [59, 27] — besides localization and mapping, prediction is also a central part of our spatial perception. Our mental model learns an abstract spatiotemporal representation of the world and efficiently predicts the future state based on the representation. This is often called *World Models*. The concept of a World Model was first discussed in the field of psychology [18] and neuroscience [24]. The idea was later introduced to the field of machine learning as a distilled rep-



(a) Ego-centric World Model [David Ha and Schmidhuber, NeurIPS'18] (b) Object-oriented World Model [Z.Lin et al. ICML'20] (c) Pedestrian World Model Nishimura et al.

Figure 1.3: Concept of Pedestrian World Model. (a) Ego-centric World Model consisting of a vison module (V) that encodes the whole image into the latent representation and memory module (M) that predict the next states in it. (b) Object-oriented World Model that performs object-wise encoding. (c) Pedestrian World Model that is characterized by an object-wise encoding and geometric memory module (GM) that predicts on-ground future states from ego-centric inputs.

resentation of the world [27]. To avoid directly model spatiotemporal changes of the environment in its high-dimensional space, the authors first embed the whole state observation into learned low-dimensional space and then predict the future state in it. By learning a transition model for an environment in low-dimensional spaces, world models allow us to explore the environment efficiently without extensive sampling of the real-world [28, 30].

By seeing view birdification as a mental model of the observer in a crowd, an object-oriented representation can be seen as an abstraction of the world. Contrary to conventional world models that predict the future state of the whole image in an ego view [27], we explore to construct a world model that can estimate the future states of each pedestrian in an object-oriented fashion, while learning interactions between them. We define such an object-oriented world model as the *Pedestrian World Model*, a transition model of the crowded environment that can continuously localize and predict the movements of all people visible to the observer. There exist a few works that consider object-wise encoding, which are referred to as object-oriented World Model [54] or Structured World Model [44]. These works, however, assume a static observer and learn the transition model

of each object from a distant and static viewpoint. As illustrated in Fig. 1.3, we are the first to model how pedestrians move conditioned on the observer's movements, *i.e.,* action. In contrast to the view birdification which geometrically reconstructs on-ground pedestrian trajectories while estimating the camera ego-motion, pedestrian world model encapsulates a transition model that predicts the future states of pedestrians conditioned by the observer's movements.

1.4 Transformers for Geometric Translation and Object Interaction Modeling

Technically, localization and prediction from in-crowd views entail three common challenges. First, estimating the motion model of pedestrians is critical as it provides the fundamental constraints by the dynamic reference points. Second, although the motion model to be estimated is described on the ground plane, the observed movements are in the 2D image. We thus need to resolve geometric 2D-to-2D reconstruction between ego-centric and on-ground views while estimating the underlying crowd motion model on the ground. Lastly, most important, the observation camera is also moving in the crowd. We need to decouple the observer's ego-motion and on-ground pedestrian movements from in-crowd views. These three problems are deeply intertwined with each other and impossible to solve independently. While many well-established techniques solve simultaneous recovery of geometric translation and the motion of dynamic objects, *i.e.,* Bird's-Eye-View (BEV) Transformation in dynamic environments [35], to our knowledge, none of past works resolve these triplets of fundamental challenges simultaneously. In this dissertation, we establish a computational model to disentangle these effectively and efficiently.

We develop a novel Transformer-based architecture based on several key insights. The model needs to learn the geometric translation between the ego-centric and the on-ground viewpoint while also learning interactions between pedestrians and an observer. The model needs to be fundamentally object-centric, which must model each pedestrian as an independent object including an observer. Finally, the model must handle a varying number of pedestrians entering in and leaving out across the frames. To satisfy all these requirements, we base our model on a Transformer [85], which is best suited for simultaneous learning of the geometric transformation and pedestrian interactions in an object-oriented

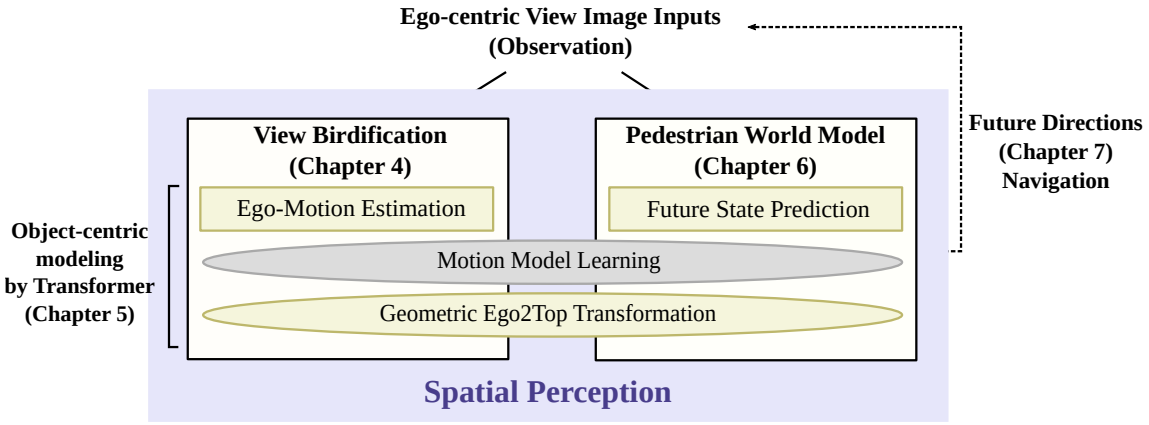


Figure 1.4: Summary of Contributions. We lay the foundation of on-ground pedestrian movement estimation and prediction from ego-centric in-crowd views. Our work can be used for various downstream applications such as navigating mobile robots in a crowd. The learned ego-motion estimator and world models offer a versatile state representation for in-crowd observations.

fashion. A primal component of the Transformer is attention mechanism [85], which adaptively applies learnable attention weights with respect to input sequences, *i.e.*, tokens. This simple yet powerful framework has achieved remarkable success across a wide range of tasks in deep learning such as language modeling [85, 43, 75], image recognition [32, 19], and set-to-set translations [48, 25], where they calculate attention over tokens,*i.e.*, embeddings of words or image patches with auxiliary positional information [92, 19]. In contrast to prior applications of Transformers which just use positional information as auxiliary cues for individual tokens, in our problem, tokens themselves have a position and velocity. In this dissertation, we pioneer the application of Transformers as a foundation architecture for both view birdification and pedestrian world model.

1.5 Contributions

As summarized in Fig. 1.4, the primary aim of this dissertation is to propose a new framework for localization and prediction in highly dynamic environments such as crowded streets. The key research question in this dissertation asks how we can achieve localization and prediction just from dynamic keypoints. Specifically, we explore the following three topics on view birdification.

Chapter 4: View Birdification from a Bayesian Perspective We first introduce view birdification, the task of recovering ground-plane movements of people in a crowd from an ego-centric video captured from an observer (e.g., a person or a vehicle) also moving in the crowd. Recovered ground-plane movements would provide a sound basis for situational understanding and benefit downstream applications in computer vision and robotics. We formulate view birdification as a geometric trajectory reconstruction problem and derive a cascaded optimization method from a Bayesian perspective. The method first estimates the observer's movement and then localizes surrounding pedestrians for each frame while taking into account the local interactions between them. We introduce three datasets by leveraging synthetic and real trajectories of people in crowds and evaluate the effectiveness of our method. The results demonstrate the accuracy of our method and set the ground for further studies of view birdification as an important but challenging visual understanding problem.

Chapter 5: Learning to recover ground-plane crowd trajectories and ego-motion This chapter introduces a novel learning-based method for view birdification. The view birdification is challenging mainly for two reasons; i) absolute trajectories of pedestrians are entangled with the movement of the observer which needs to be decoupled from their observed relative movements in the ego-centric video, and ii) a crowd motion model describing the pedestrian movement interactions is specific to the scene yet unknown a priori. For this, we introduce a Transformer-based network referred to as ViewBirdiformer which implicitly models the crowd motion through self-attention and decomposes relative 2D movement observations onto the ground-plane trajectories of the crowd and the camera through cross-attention between views. Most important, ViewBirdiformer achieves view birdification in a single forward pass which opens the door to accurate real-time, always-on situational awareness.

Chapter 6: Pedestrian World Model In this chapter, we explore the view birdification from an aspect of the ego-centric world model. We reformulate the birdification as a prediction model conditioned on the observer's movement. We also show that a Transformer-based architecture is best suited for simultaneous learning of the interaction between pedestrians and geometric 2D-to-2D transformation. To our knowledge, our work is the first to construct an object-oriented world model considering the observer's action, *i.e.*, movements.

1.6 Dissertation Overview

The rest of this dissertation is organized as follows.

Chapter 2 reviews previous work relevant to localization and prediction in dynamic environments.

Chapter 3 formulates view birdification as a novel computer vision task and preliminary notations commonly referred to in the following three chapters (**Chapters 4–6**).

Chapter 4 proposes a Bayesian formulation of view birdification in a cascaded optimization approach. The content covered in this chapter is published in the M.Nishimura et al. "View Birdification: Ground-Plane Localization from Perceived Movements", in Proceedings of the British Machine Vision Conference (BMVC), 2021.

Chapter 5 presents a data-driven approach to view birdification based on a Transformer architecture. The content covered in this chapter is published in the M.Nishimura et al. "ViewBirdiformer: Learning to recover ground-plane crowd trajectories and ego-motion from a single ego-centric view" in the IEEE/RAS Robotics and Automation Letters (RA-L), 2022.

Chapter 6 extends view birdification as a World Model. This chapter contains unpublished work entitled M.Nishimura et al., "InCrowdFormer: On-Ground Pedestrian World Model From Egocentric Views".

Chapter 7 summarizes the dissertation and discusses possible future directions.

Chapter 2

Related Work

View birdification and Pedestrian World Model are related to a number of computer vision and robotics problems, which we review in this chapter. Table 2.1 summarizes the differences between view birdification and its relevant works. Table 2.2 also compares Pedestrian World Model with prior work.

2.1 Bird's Eye View Transformation

As summarized in Table 2.1, View Birdification [66] is not the same as bird's-eye view (BEV) transformation [96, 35, 100, 77]. BEV transformation refers to the task of rendering a 2D top-down view image from an on-ground ego-centric view and concerns the appearance of the surroundings as seen from the top and does not resolve the ego-motion, *i.e.,* all recovered BEVs are still relative to the observer. View birdification, in contrast, reconstructs both the observer's and surrounding pedestrians' locations on the ground so that the relative movements captured in the ego-centric view can be analyzed in a single world coordinate frame on the ground (*i.e.,* "birdified"). View birdification thus fundamentally differs from BEV transform as it is inherently a 3D transform that accounts for the ego-motion, *i.e.,* the 2D projections of surrounding people in the 2D ego-view need to be implicitly or explicitly lifted into 3D and translated to cancel out the jointly estimated ego-motion of the observer before being projected down onto the ground-plane. Nishimura *et al.* introduced a geometric method for view birdification [66], which explicitly transforms the 2D projected pedestrian movements into 3D but on the ground plane with a graph energy minimization by leveraging analytically expressible crowd motion models [33]. Our method fundamentally differs from this in that the transformation from 2D in-image movement to on-ground motion

Task	Target	Input		Output		Object Density	
		static	dynamic	ego-motion	object location	sparse	dense
BEV Transformation [49]	image, objects	✔			✔	✔	✔
3D Object Detection [52, 53]	objects	✔			✔	✔	✔
3D Multi-Object Tracking [36]	objects		✔		✔	✔	✔
Dynamic SLAM [37]	keypoints	✔	✔	✔	✔	✔	
View Birdification [70] (Ours)	objects		✔	✔	✔	✔	✔

Table 2.1: View Birdification is the problem of recovering ground-plane movements of people in a crowd from an ego-centric video captured from an observer. While none of the other tasks recover the absolute layout of dynamic objects from their observations, view birdification achieves simultaneous recovery of the absolute trajectories including the observer ego-motion even in a dense crowd only from their perceived movements relative to an observer.

as well as the on-ground coordination of pedestrian motion is jointly learned from data.

Transforming first-person-view (FPV) images into top-down maps (BEV) images has become important for autonomous driving [77, 100, 96]. Most works directly learn the frame-by-frame mapping in a data-driven fashion without taking into account ego-motion [8, 99]. In other words, they are only relative to the observer. Nishimura et al. recently proposed view birdification which is the task of reconstructing on-ground positions and trajectories of pedestrians and the observer, *i.e.*, , simultaneous decoupling of ego-motion from observed 2D motions in the FPV and anchoring of all motions in absolute coordinates [66]. Our work differs from BEV transform and view birdification in two critical points. First, we aim at constructing a Pedestrian World Model not from the appearance, but from the movements, which results in a compact and efficient representation that can easily generalize. Second, unlike previous approaches which only perform deterministic mappings [77], we derive a probabilistic formulation that can handle uncertainty underlying object distances in the FPV.

2.2 Dynamic SLAM

Dynamic SLAM and its variants inherently rely on the assumption that the world is static [65, 21]. Dynamic objects cause feature points to drift and contaminate the ego-motion estimate and consequently the 3D reconstruction. Past methods have made SLAM applicable to dynamic scenes, "despite" these dynamic objects, by

treating them as outliers [31] or explicitly tracking and filtering them [9, 97, 86, 6]. A notable exception is Dynamic Object SLAM which explicitly incorporates such objects into its geometric optimization [37, 95, 34]. The method detects and tracks dynamic objects together with static keypoints, but assumes that the dynamic objects in view are rigid and obey a simple motion model that results in smoothly changing poses. None of the above methods consider the complex pedestrian interactions in the crowd [33, 84, 26, 98]. Our method fundamentally differs from dynamic SLAM in that it reconstructs both the observer's ego-motion and the on-ground trajectories of surrounding dynamic objects without relying on any static key-point, while also recovering the interaction between surrounding dynamic objects. In other words, the movements themselves are the features.

2.3 Crowd Modeling

Modeling human behavior in crowds is essential for a wide range of applications including crowd simulation [50], trajectory forecasting [2, 39, 26], and robotic navigation [69, 83, 3]. Popular approaches include multi-agent interactions based on social force models [33, 60, 3], reciprocal force models [84], and imitation learning [83]. Recently, data-driven approaches have achieved significant performance gains on public crowd datasets [2, 26, 39]. All these approaches, however, are only applicable to near top-down views. Forecasting the future location of people from first-person viewpoints has also been explored [94, 57], but they are limited to localization in the image plane. View birdification may provide a useful foundation for these crowd modeling tasks.

2.4 Non-rigid Structure from Motion

Reconstruction of point trajectories is also studied in the literature on non-rigid structure from motion (NRSfM) [40, 78], in particular as multi-body [47] and trajectory-based [1, 72] approaches. NRSfM exploits the inherent global dynamic structure embodied by the target surface and the camera motion. In contrast, our focus is pedestrian trajectories that interact locally in an on-the-fly manner and do not exhibit coherent global structures that we can leverage.

Method	state	ego2top	action	V	M
RSSM [27]	image	–	✓	CNN	GRU
TSSM [14, 20]	image	–	✓	CNN	Transformer
C-SWM [44]	object	–	–	CNN	GNN
G-SWM [54]	object	–	–	CNN	RNN
Ours	object	✓	✓	Transformer	

Table 2.2: We introduce, to our knowledge, the first egocentric crowd world model that models pedestrian movements solely from their in-crowd egocentric observations with a unified Transformer architecture that embodies ego2top transform and dependency of pedestrian movements on the observer's actions.

2.5 3D Multi-Object Tracking (MOT)

3D MOT concerns the detection and tracking of target objects in a video sequence while estimating their 3D locations on the ground [81, 55]. Most recent works aim to improve tracklet association across frames [55]. These approaches, however, assume a simple motion model independent of the camera ego-motion [91], which hardly applies to a dynamic observer in a crowd with complex interactions with other pedestrians. 3D MOT in a video with a dynamic observer [36] has been studied, but the observer motion is known from an external GPS which is often inaccurate [38]. Our work focuses on reconstructing both the observer ego-motion and surrounding pedestrians in a crowd, while simultaneously learning their complex interactions, which complements these works for visual situational awareness and surveillance.

2.6 World Models

A world model is an abstract representation of our environment and its transitions [41]. Ha and Schmidhuber recently introduced the idea of building a world model with a perception model (V) and a transition model (M), so that a simple agent controller (C) [27] can be learned in the world model with generated rollouts of simulated experiences [29, 28, 30, 20]. Most such world models encode the whole image into a latent code as the environment representation. Structured World Models [44] and Object-centric World Models [54] learn object-oriented representations about the world (*i.e.*, , individual objects constitute the environment representation). Even in these models, however, the observer never inter-

acts with the environment and object transitions are simply observed from static viewpoints. In sharp contrast, we are interested in modeling a human-populated environment with an object-centric world model that also contains the dynamic observer itself.

2.7 Crowd Navigation

Earlier works used analytical crowd motion models [33, 84] to predict future pedestrian locations for navigating a robot in a crowd. Deep reinforcement learning (DRL) approaches often incorporate learning-based crowd models such as social pooling [13, 2] and graph neural networks [12, 16, 39]. Most works, however, assume that they can perfectly observe ground-truth positions and velocities of pedestrians directly in a top-down view, at every timestep, which is hardly plausible in the real world. Few works tackle ego-centric view navigation, *i.e.*, , path planning directly from an observer's ego-centric view in the crowd. Dugas *et al.* [20] constructed synthetically generated human-populated environments with a game-engine for vision-based navigation. The domain gap between real and synthetic environments is, however, not negligible both in appearances and pedestrian movements. In contrast, our egocentric pedestrian world model is independent of pedestrian appearance and our dataset can be easily augmented with arbitrary combinations of observer actions and real crowd trajectories. We show that this enables direct application of our model to unseen real video sequences.

Chapter 3

View Birdification

This chapter provides preliminary knowledge and concepts that are commonly used in subsequent chapters.

3.1 Background

We as human beings are capable of mentally visualizing our surroundings from a third-person view. Imagine walking down a street alongside other pedestrians. Your mental model of the surrounding movements of people is not a purely two-dimensional one, but rather in 3D, albeit imperfect, in which you can virtually fly around. It lets you anticipate potential collisions so that you can avoid them or guess the goal of another person so that you can follow. Some people have exceptionally high capabilities in forming such a virtual view (*e.g.*, a professional soccer player), but nonetheless, we all rely on this 3D spatial sense to complement our ego-centric view in our daily lives. Endowing such global 3D spatial perception with computers, however, remains elusive. Despite the significant progress in computational 3D and motion perception, including stereo, structure from motion, and optical flow estimation, a bottom-up approach of first reconstructing the 3D geometry and motion and then changing the viewpoint would be brittle. Its success would inherently hinge on the accuracy of each step which is prone to fundamental ambiguities between them. Can we bypass these and directly obtain a virtual perspective of the surroundings? More specifically, can we recover the dynamically changing global layout of people moving around ourselves solely from images captured from our vantage point while we also move around?

In this chapter, we introduce *view birdification* in a crowd [70], the problem of computing a bird-eye's view of the movements of surrounding people from

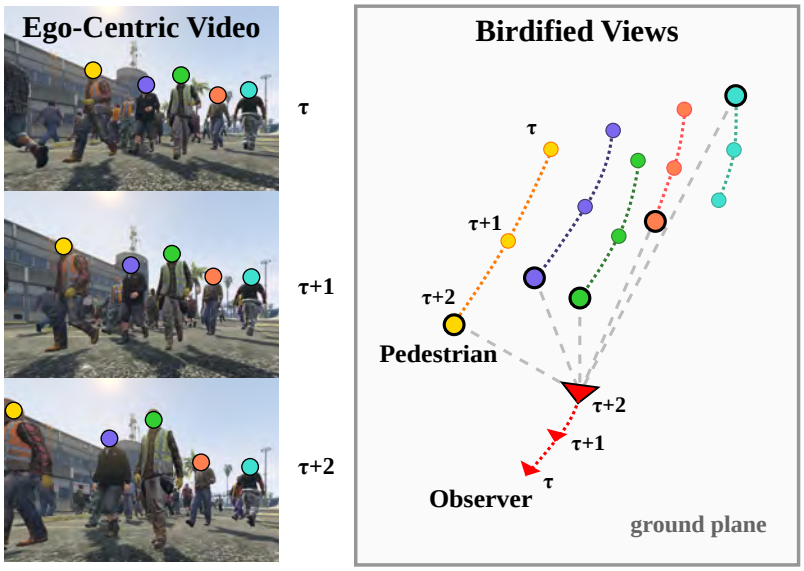


Figure 3.1: We introduce view birdification of a crowd, the task of estimating the movements of surrounding people on the ground plane (right) from a single dynamic ego-centric image sequence (left), and derive a stratified optimization method based on the geometric relations of pedestrians' projections and interactions.

a single ego-centric view of a moving person (see Fig. 3.1) and derive a geometric solution to it. View birdification differs from recent works on bird-eye view rendering where the goal is to render a bird's eye view image from a given ego-centric image, *i.e.*, view transformation such that the scene appearance is imaged fronto-parallel to the ground. We are, in contrast, interested in deciphering and laying out the movements of people on the ground plane from a temporal sequence of a dynamic ego-centric view. In other words, from a personal perspective of a crowd, we would like to see how everybody is moving (not how they look) as seen from the top (*i.e.*, a bird hovering over the crowd).

View birdification of a crowd would have a wide range of applications. It will let us analyze the global and local interactions of people from a holistic perspective both in space and time, which would benefit areas such as navigation [69, 3], movement prediction [39, 26, 2], and surveillance [45]. It can also offer a crucial visual perception for self-driving cars to gauge surrounding activities.

Unlike bird-eye view rendering which can be formulated as an image-to-image transformation (*e.g.*, with a deep neural network), view birdification does not concern the appearance of the scene captured in the ego-centric view. From observations of dynamically moving objects, our method localizes the moving

camera and simultaneously maps the dynamic objects on the ground plane. This is reminiscent of SLAM but with the fundamental difference that everything is dynamic. The dynamic objects (*i.e.*, people) also do not embody any low-dimensional structure as often assumed in non-rigid structure from motion.

View birdificaction is based on two key insights. First, the movements of dynamic keypoints (*e.g.*, head-points of pedestrians) are not arbitrary, but exhibit coordinated motion that can be expressed with crowd flow models [33, 73]. That is, the interaction of pedestrians' movements in a crowd can be locally described with analytic or data-driven models. Second, the scale and difference of human heights are proportional to estimated geometric depth [56]. In other words, the positions of pedestrians on the ground plane can be constrained along the lines that pass through a center of projection. These insights lend us a natural formulation of view birdification as a geometric reconstruction problem.

3.2 Geometric View Birdification

A typical scenario for view birdification is when a person with a body-worn camera is immersed in a crowd consisting of people heading toward their destinations while implicitly interacting with each other. Our goal is to deduce the global movements of people from the local observations in the ego-centric video captured by a single person.

3.2.1 Problem Setting

As a general setup, we assume that K people are walking on a ground plane and an observation camera is mounted on one of them. We set the z-axis of the world coordinate system to the normal of the ground plane (x-y plane) and denote the on-ground location of the k^{th} pedestrian as $\boldsymbol{x}_k = [x_k, y_k]^\top$. Let us denote the location of 0^{th} person in the crowd $\boldsymbol{x}_0$ as the observer capturing the ego-centric video of pedestrians $k \in \{1, 2, \ldots, K\}$ who are visible to the observer. The observation camera is located at $[x_0, y_0, h_0]^\top$, where the mounted height h_0 is constant across the frames. We assume that the viewing direction is parallel to the ground plane, *e.g.*, the person has a camera mounted on the shoulder. The same assumption applies when the observer is a vehicle or a mobile robot. At each timestep τ, the pedestrians are observed by a camera with the pose $[R^\tau | \boldsymbol{t}^\tau]$ consisting of rotation matrix R and translation vector $\boldsymbol{t}$. Assuming that the viewing direction of the

camera is stabilized and parallel to the ground plane, we can approximate the rotation angles about the x- and y-axis to be 0 across the frames. That is, the camera pose to be estimated is represented by its 2D rotation about z-axis $R(\Delta\theta_z) \in SO(2)$ and 2D translation $\boldsymbol{t} = -R(\Delta\theta_z)\Delta\boldsymbol{x}_0 \in \mathbb{R}^2$ on the x-y plane.

We assume that the bounding boxes of the people captured in the ego-video are already extracted. For this, we can use an off-the-shelf multi-object tracker [93, 90] which provides the state of each pedestrian on the image plane $\boldsymbol{s}_k^\tau = \left[u_k^\tau, v_k^\tau, l_k^\tau\right]^\top$ which consists of the projections of center location and height, $\boldsymbol{p}_k^\tau = \left[u_k^\tau, v_k^\tau\right]^\top$ and l_k^τ, respectively. Note that our method is agnostic to the actual tracking algorithm. Pedestrian IDs $k \in \{1, 2, \ldots, K\}$ can also be assigned by the tracker. Given a sequence of pedestrian states $\mathcal{S}_k$ from the first visible frame τ_1 to the last visible frame τ_2, *i.e.*, , $\mathcal{S}_k^{\tau_1:\tau_2} = \{\boldsymbol{s}_k^{\tau_1}, \boldsymbol{s}_k^{\tau_1+1}, \ldots, \boldsymbol{s}_k^{\tau_2}\}$, our goal is to simultaneously reconstruct the K trajectories of the surrounding pedestrians $\mathcal{X}_k^{\tau_1:\tau_2} = \{\boldsymbol{x}_k^{\tau_1}, \boldsymbol{x}_k^{\tau_1+1}, \ldots, \boldsymbol{x}_k^{\tau_2}\}$ and that of the observation camera $\mathcal{X}_0^{\tau_1:\tau_2} = \{\boldsymbol{x}_0^{\tau_1}, \boldsymbol{x}_0^{\tau_1+1}, \ldots, \boldsymbol{x}_0^{\tau_2}\}$ with its viewing direction $\mathcal{R}^{\tau_1:\tau_2} = \{R^{\tau_1}, R^{\tau_1+1}, \ldots, R^{\tau_2}\}$ on the ground plane.

3.2.2 Geometric Observation Model

We assume a regular perspective ego-centric view or a 360° cylindrical projection view. The following derivation also applies to other linear projection models including generic quasi-central cameras for fish-eye lens [10].

Perspective Projection Model In the case of perspective projection with focal length f and intrinsic matrix $A \in \mathbb{R}^{3\times3}$, the distance of the pedestrian from the observer is proportional to the ratio of the pedestrian height h_k and its projection l_k, *i.e.*, h_k/l_k. Given the center projection of the pedestrian in the image plane $\boldsymbol{s}_k = [u_k, v_k, l_k]$, the on-ground location estimate of the pedestrian relative to the camera $z_k = [\tilde{x}_k, \tilde{y}_k, 0]^\top$ can be computed by inverse projection of the observed image coordinates,

$$\begin{bmatrix} \tilde{x}_k & \frac{h_k}{2} & \tilde{y}_k \end{bmatrix}^\top = \frac{f h_k}{l_k} A^{-1} \begin{bmatrix} u_k & v_k & 1 \end{bmatrix}^\top , \tag{3.1}$$

where the intrinsic A and focal length f are known since the observation camera can be calibrated a priori. The relative coordinates z_k are thus scaled by the unknown pedestrian height parameter h_k.

Cylindrical Projection Model Mobile platforms often use a 360° panorama view for a full view of the surroundings, which are composed of synchronized RGB sensor images. Given a stitched 360° cylindrical image with image width W and the observed pedestrian state $\boldsymbol{s}_k = [u_k, v_k, l_k]^\top$ in the image, the location angle ϕ [rad] for the pedestiran position $\boldsymbol{p}_k = [u_k, v_k]^\top$ on the cylinder circle becomes

$$\phi = 2\pi \frac{u_k}{W} - \pi. \tag{3.2}$$

The inverse projection depth from the center of the circle $\tilde{r}_k$ is proportional to the ratio of the pedestrian height h_k and its projection l_k,

$$\tilde{r}_k = \tilde{y}_k \sec(\phi) = f \frac{h_k}{l_k}. \tag{3.3}$$

The on-ground location estimates of the pedestrian can be recovered as $\boldsymbol{z}_k = [\tilde{r}_k \sin(\phi), \tilde{r}_k \cos(\phi), 0]^\top$.

The absolute position of the pedestrian $\boldsymbol{x}_k = [x_k, y_k]^\top$ can be computed by the relative coordinates $\boldsymbol{z}_k = [\tilde{x}_k, \tilde{y}_k]^\top$, the camera position $\boldsymbol{x}_0 = [x_0, y_0]^\top$, and the viewing direction θ_z about z-axis,

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = R_z(\theta_z)^\top \begin{bmatrix} \tilde{x}_k \\ \tilde{y}_k \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}. \tag{3.4}$$

In what follows, we assume the most general case, *i.e.*, perspective projection. The optimization pipeline, however, can be applied to any type of linear projection model without major changes.

Chapter 4

View Birdification from a Bayesian Perspective

In this chapter, we derive a solution for geometric view birdification based on cascaded optimization. Our stratified optimization consists of the observer's camera ego-motion estimation with pedestrian movement interactions as pairwise constraints and pedestrian localization given the ego-motion estimate and height priors on the pedestrians. We first solve this camera ego-motion estimation by gradient descent and then localize each pedestrian given the observer's camera position as a combinational optimization problem with pairwise interaction constraints.

We experimentally validate our method on both synthetic and real trajectories extracted from publicly available crowd datasets. We create a photorealistic crowd dataset that simulates real camera projection with a limited field of view and occluded pedestrian observations while moving in the crowd. These datasets allow us to quantitatively evaluate our method systematically. Experimental results demonstrate the effectiveness of our approach for view birdification in crowds of various densities. The results on the photorealistic crowd dataset show the end-to-end effectiveness of our method, from person detection to localization on the ground plane, demonstrating its performance in real-world use. We also test our method on real-robot dataset captured in crowds. The results show that our method can work both for body-worn cameras and mobile robot platforms. We believe these results have strong implications in computer vision and robotics as they establish view birdification as a foundation for downstream visual understanding applications including crowd behavior analysis and robot navigation.

4.1 A Cascaded Optimization for View Birdification

In this section, we introduce a cascaded optimization approach to the geometric view birdification problem based on a Bayesian perspective. We first describe the overall energy minimization framework and then derive energy functions to be optimized for the two typical models.

4.1.1 A Bayesian Formulation

When a frame is pre-processed to a set of states $\mathcal{S}_{1:K}^{\tau} = \{\boldsymbol{s}_1^{\tau}, \boldsymbol{s}_2^{\tau}, \ldots, \boldsymbol{s}_K^{\tau}\} \in \mathbb{R}^{3\times K}$ at time τ, we obtain a set of on-ground position estimates relative to a camera $\mathcal{Z}_{1:K}^{\tau} = \{\boldsymbol{z}_1^{\tau}, \boldsymbol{z}_2^{\tau}, \ldots, \boldsymbol{z}_K^{\tau}\} \in \mathbb{R}^{2\times K}$ corresponding to the states $\mathcal{S}_{1:K}^{\tau}$. Assuming that we have sequentially estimated on-ground positions up to time $\tau-1$, $\mathcal{X}_{0:K}^{\tau_0:\tau-1} = \{\mathcal{X}_0^{\tau_0:\tau-1}, \mathcal{X}_1^{\tau_0:\tau-1}, \ldots, \mathcal{X}_K^{\tau_0:\tau-1}\} \in \mathbb{R}^{2\times(K+1)\times\Delta\tau}$ with a temporal window of $\Delta\tau$ and its initial timestamp $\tau_0 = \tau - \Delta\tau$, the posterior probability of the on-ground positions $\mathcal{X}_{0:K}^{\tau} = \{\boldsymbol{x}_0^{\tau}, \boldsymbol{x}_1^{\tau}, \ldots, \boldsymbol{x}_K^{\tau}\} \in \mathbb{R}^{2\times(K+1)}$ at time τ can be factorized as

$$\begin{aligned} &p(\mathcal{X}_{0:K}^{\tau}|\mathcal{Z}_{1:K}^{\tau}, \mathcal{X}_{0:K}^{\tau_0:\tau-1}) \\ &\propto p(\mathcal{X}_{0:K}^{\tau}|\mathcal{X}_{0:K}^{\tau_0:\tau-1}) p(\mathcal{Z}_{1:K}^{\tau}|\mathcal{X}_{0:K}^{\tau}, \mathcal{X}_{0:K}^{\tau_0:\tau-1}) \,. \end{aligned} \tag{4.1}$$

Let $\Delta\boldsymbol{x}_0^{\tau} = [\Delta x_0^{\tau}, \Delta y_0^{\tau}, \Delta\theta_z^{\tau}] \in \mathbb{R}^3$ be the camera ego-motion from timestep $\tau-1$ to τ consisting of a 2D translation $[\Delta x_0, \Delta y_0]^{\top}$ and a change in viewing direction $\Delta\theta_z$ on the ground plane (x-y plane). The optimal motion of the camera $\Delta\hat{\boldsymbol{x}}_0^{\tau}$ and those of the pedestrians $\hat{\mathcal{X}}_{1:K}^{\tau} = \{\boldsymbol{x}_1^{\tau}, \boldsymbol{x}_2^{\tau}, \ldots, \boldsymbol{x}_K^{\tau}\} \in \mathbb{R}^{2\times K}$ can be estimated as those that maximize the posterior distribution (Eq. (4.1)). The motion of observed pedestrians $\mathcal{X}_{1:K}^{\tau-1:\tau}$ are strictly constrained by the observing camera position $\boldsymbol{x}_0^{\tau}$ and its viewing direction θ_z^{τ}. With recovered pedestrian parameters $\hat{\mathcal{X}}_{1:K}^{\tau}$, the optimal estimate of the camera ego-motion $\Delta\hat{\boldsymbol{x}}_0^{\tau}$ becomes

$$\begin{aligned} \Delta\hat{\boldsymbol{x}}_0^{\tau} = &\underset{\Delta\boldsymbol{x}_0^{\tau}\in\mathbb{R}^3}{\operatorname{argmax}} \, p(\boldsymbol{x}_0^{\tau}|\mathcal{X}_0^{\tau_0:\tau-1}) \\ &\prod_k p(\boldsymbol{x}_k^{\tau}|\hat{\mathcal{X}}_k^{\tau_0:\tau-1}, \Delta\boldsymbol{x}_0^{\tau}) p(\boldsymbol{z}_k^{\tau}|\boldsymbol{x}_k^{\tau}, \Delta\boldsymbol{x}_0^{\tau}) \,, \end{aligned} \tag{4.2}$$

where $p(\boldsymbol{x}_0^{\tau}|\mathcal{X}_0^{\tau_0:\tau-1})$ and $p(\boldsymbol{x}_k^{\tau}|\mathcal{X}_k^{\tau_0:\tau-1}, \Delta\boldsymbol{x}_0^{\tau})$ are motion priors of the camera and pedestrians conditioned on the camera motion, respectively. If the observer camera is mounted on a pedestrian following the crowd flow, $p(\boldsymbol{x}_0^{\tau}|\mathcal{X}_0^{\tau_0:\tau-1})$ obeys the same motion model as $p(\boldsymbol{x}_k^{\tau}|\mathcal{X}_k^{\tau_0:\tau-1})$.

As in previous work for pedestrian detection [56], we assume that the heights of pedestrians h_k follow a Gaussian distribution. This lets us define the likelihood of observed pedestrian positions $\boldsymbol{z}_k^\tau$ relative to the camera $\boldsymbol{x}_0^\tau$ as

$$\boldsymbol{z}_k^\tau \sim p(\boldsymbol{z}_k^\tau | \boldsymbol{x}_k^\tau; h_k) = \mathcal{N}(\mu_h, \sigma_h^2)\,, \tag{4.3}$$

where $\mathcal{N}(\mu_h, \sigma_h^2)$ is a Gaussian distribution with mean μ_h and variance σ_h^2. Once the ego-motion of the observing camera is estimated as $\Delta\hat{\boldsymbol{x}}_0^\tau$, the pedestrian positions $\hat{\mathcal{X}}_{1:K}^\tau$ that maximize the posterior $p(\mathcal{X}_{1:K}^\tau | \mathcal{Z}_{1:K}^\tau, \Delta\boldsymbol{x}_0^\tau)$ can be obtained as

$$\hat{\mathcal{X}}_{1:K}^\tau = \underset{\boldsymbol{x}_k^\tau \in \mathcal{X}_{1:K}^\tau}{\operatorname{argmax}} \prod_k p(\boldsymbol{x}_k^\tau | \mathcal{X}_k^{\tau_0:\tau-1}, \Delta\hat{\boldsymbol{x}}_0^\tau) p(\boldsymbol{z}_k^\tau | \boldsymbol{x}_k^\tau, \Delta\hat{\boldsymbol{x}}_0^\tau)\,. \tag{4.4}$$

That is, we can estimate the ego-motion of the observer constrained by the perceived pedestrian movements which conform to the crowd motion prior and the observation model.

When the camera observes a large number of pedestrians that conforms to a known crowd motion model, regardless of whether the camera motion is consistent with dominant crowd flow, the camera ego-motion estimates depend heavily on the observed crowd movements and are less sensitive to the assumed ego-motion model. In such cases, Eq. (4.2) can be re-written as

$$\Delta\hat{\boldsymbol{x}}_0^\tau = \underset{\Delta\boldsymbol{x}_0^\tau \in \mathbb{R}^3}{\operatorname{argmax}} \prod_{k=1}^{K} p(\boldsymbol{x}_k^\tau | \hat{\mathcal{X}}_k^{\tau_0:\tau-1}, \Delta\boldsymbol{x}_0^\tau) p(\boldsymbol{z}_k^\tau | \boldsymbol{x}_k^\tau, \Delta\boldsymbol{x}_0^\tau)\,. \tag{4.5}$$

As long as the camera observes a sufficient number of pedestrians walking in diverse directions, our method can successfully birdify its view.

4.1.2 Energy Minimization

Once the camera ego-motion is estimated, we can update the individual locations of pedestrians given the ego-motion in an iterative refinement process. View birdification can thus be solved with a cascaded optimization which first estimates the camera ego-motion and then recovers the relative locations between the camera and the pedestrians given the ego-motion estimate while taking into account the local interactions between pedestrians. Minimization of the negative

log probabilities, Eqs. (4.2) and (4.4), can be expressed as

$$\underset{\Delta \boldsymbol{x}_0^\tau \in \mathbb{R}^3}{\text{minimize}} \;\; \mathcal{E}_c(\Delta \boldsymbol{x}_0^\tau; \hat{\mathcal{X}}_{1:K}^\tau, \mathcal{Z}_{1:K}^\tau, \mathcal{X}_{0:K}^{\tau_0:\tau-1}) \,, \tag{4.6}$$

$$\begin{aligned} &\text{subject to} \\ &\hat{\mathcal{X}}_{1:K}^\tau = \underset{\mathcal{X}_{1:K}^\tau}{\operatorname{argmin}} \, \mathcal{E}_p(\mathcal{X}_{1:K}^\tau; \Delta \hat{\boldsymbol{x}}_0^\tau, \mathcal{Z}_{1:K}^\tau, \mathcal{X}_{0:K}^{\tau_0:\tau-1}) \,, \end{aligned} \tag{4.7}$$

where we define the energy functions for positions of camera $\mathcal{E}_c$ and pedestrians $\mathcal{E}_c$ as

$$\mathcal{E}_c(\Delta \boldsymbol{x}_0^\tau; \hat{\mathcal{X}}_{1:K}^\tau, \mathcal{Z}_{1:K}^\tau, \mathcal{X}_{1:K}^{\tau_0:\tau-1}) = -\ln p(\boldsymbol{x}_0^\tau | \mathcal{X}_0^{\tau_0:\tau-1}) + \mathcal{E}_p \,, \tag{4.8}$$

$$\begin{aligned} &\mathcal{E}_p(\mathcal{X}_{1:K}^\tau; \Delta \hat{\boldsymbol{x}}_0^\tau, \mathcal{Z}_{1:K}^\tau, \mathcal{X}_{0:K}^{\tau_0:\tau-1}) \\ &= \sum_{k=1}^{K} -\ln p(\boldsymbol{x}_k^\tau | \mathcal{X}_k^{\tau_0:\tau-1}, \Delta \boldsymbol{x}_0^\tau) + \sum_{k=1}^{K} -\ln p(\boldsymbol{z}_k^\tau | \boldsymbol{x}_k^\tau, \Delta \boldsymbol{x}_0^\tau) \,. \end{aligned} \tag{4.9}$$

We minimize the energy in Eq. (4.6) by first computing an optimal camera position $\hat{\boldsymbol{x}}_0^\tau$ from Eq. (4.6) with gradient descent and initial state $\boldsymbol{x}_0^\tau = \boldsymbol{x}_0^{\tau-1}$. Given the estimate of the observer location $\hat{\boldsymbol{x}}_0^\tau$, we then estimate the pedestrian locations by solving the combinatorial optimization problem in Eq. (4.7) for $\mathcal{X}_k^\tau$ while considering all possible combinations of $\{\boldsymbol{x}_1^\tau, \ldots, \boldsymbol{x}_K^\tau\}$ under the projection constraint in Eq. (3.1) and the assumed pedestrian interaction model.

This can be interpreted as a fully connected graph consisting of K pedestrian nodes with unary potential and interaction edges with pairwise potential. Similar to prior works on low-level vision problems [5, 51], Eq. (4.9) can be optimized by iterative message passing [22] on the graph. The possible states $\boldsymbol{x}_i$ are uniformly sampled on the projection line around μ_h with interval $[\mu_h - \delta S/2, \mu_h + \delta S/2]$, where S is a number of samples and $\delta = 0.01$. Considering only pairwise interactions and Gaussian potential, the complexity of the optimization is $\mathcal{O}(KS^2T)$, where T is the number of iterations required for convergence. We use two types of analytical interaction models, ConstVel [79] and Social Force [33]. In what follows, we provide a detailed derivation of energy functions.

4.1.3 Pedestrian Interaction Models

We formulated view birdification as an iterative energy minimization problem that consists of a pedestrian interaction model $p(\boldsymbol{x}_k^\tau|\mathcal{X}_k^{\tau_0:\tau-1})$ and a likelihood $p(\boldsymbol{z}_k^\tau|\boldsymbol{x}_k^\tau, \Delta\boldsymbol{x}_0^\tau)$ defined by the geometric observation model with ambiguities arising from human height estimates (Eq. (4.3)). Our framework is not limited to a specific pedestrian interaction model, and any type of model that explains pedestrian interactions in a crowd can be incorporated. In the following, we consider two example models with a temporal window of $\Delta\tau = 2$.

Constant Velocity ConstVel [79] is a simple yet effective model of pedestrian interactions in a crowd which simply linearly extrapolates future trajectories from the last two frames

$$p(\boldsymbol{x}_k^\tau|\mathcal{X}_k^{\tau-2:\tau-1}) \sim \exp\left[-\|\boldsymbol{x}_k^\tau - 2\boldsymbol{x}_k^{\tau-1} + \boldsymbol{x}_k^{\tau-2}\|^2\right] . \tag{4.10}$$

The model is independent of other pedestrians and the overall pedestrian interaction model can be factorized as $p(\mathcal{X}_{1:K}^\tau|\mathcal{X}_{1:K}^{\tau-2:\tau-1}) = \prod_{k=1}^K p(\boldsymbol{x}_k^\tau|\mathcal{X}_k^{\tau-2:\tau-1})$. The energy model $\mathcal{E}_p$ is rewritten as

$$\mathcal{E}_p = \sum_{k=1}^K -\ln p(\boldsymbol{x}_k^\tau|\mathcal{X}_k^{\tau-2:\tau-1}) + \sum_{k=1}^K -\ln p(\boldsymbol{z}_k^\tau|\boldsymbol{x}_k^\tau, \Delta\boldsymbol{x}_0^\tau) . \tag{4.11}$$

Social Force The Social Force Model [33] is a well-known physics-based model that simulates multi-agent interactions with reciprocal forces, which is widely used in crowd analysis and prediction studies [60, 84]. Each pedestrian k with a mass m_k follows the velocity $d\boldsymbol{x}/dt^2$

$$m_k \frac{d^2\boldsymbol{x}_k}{dt^2} = \boldsymbol{F}_k = \boldsymbol{F}_p(\boldsymbol{x}_k) + \boldsymbol{F}_r(\mathcal{X}_\mathcal{C}) , \tag{4.12}$$

where $\boldsymbol{F}_k$ is the force on $\boldsymbol{x}_k$ consisting of the personal desired force $\boldsymbol{F}_p$ and the reciprocal force $\boldsymbol{F}_r$. The personal desired force is proportional to the discrepancy between the current velocity and that desired

$$\boldsymbol{F}_p(\boldsymbol{x}_k) = \frac{1}{\eta}\left(\boldsymbol{w}_k - \frac{d\boldsymbol{x}_k}{dt}\right) , \tag{4.13}$$

where $\boldsymbol{w}_k$ denotes the desired velocity which can be empirically approximated as the average velocity of neighboring pedestrians $i \in \mathcal{N}(\boldsymbol{x}_k)$ [60].

The form of reciprocal force $\boldsymbol{F}_r$ can be determined by the set of interactions between pedestrian nodes $\boldsymbol{x}_i \in \mathcal{X}_\mathcal{C}$. To reduce the complexity of optimization, we approximate multi-human interaction $\boldsymbol{F}_r(\mathcal{X}_\mathcal{C})$ with a collection of pairwise interactions $\boldsymbol{F}_r(\boldsymbol{x}_i, \boldsymbol{x}_k)$. We assume a standard Gaussian potential to simulate the reciprocal force between two pedestrians

$$\boldsymbol{F}_r(\boldsymbol{x}_i, \boldsymbol{x}_k) = -\nabla \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_k\|^2}{2\sigma^2} \right] \right) . \tag{4.14}$$

Without loss of generality, we omit m_k as $m_k = 1$, assuming that the mass of pedestrians in a crowd is almost consistent. Taking the last two frames as inputs, the complete pedestrian interaction model becomes

$$\begin{aligned} &p(\mathcal{X}_{1:K}^{\tau} | \mathcal{X}_{1:K}^{\tau-2:\tau-1}) \\ &\sim \prod_k \exp\left[- \left\| \boldsymbol{F}_p(\boldsymbol{x}_k^{\tau}) - \frac{d^2 \boldsymbol{x}_k^{\tau}}{dt^2} \right\| \right] \prod_{(i,k)\in\mathcal{X}_\mathcal{C}} \exp\left[- \|\boldsymbol{F}_r(\boldsymbol{x}_i^{\tau}, \boldsymbol{x}_k^{\tau})\| \right] . \end{aligned} \tag{4.15}$$

Taking negative log probabilities, the overall energy model in Eq. (4.15) becomes

$$\mathcal{E}_p = \sum_k D_k(\boldsymbol{x}_k^{\tau}; \mathcal{X}_k^{\tau-2:\tau-1}) + \sum_{(i,k)\in\mathcal{X}_\mathcal{C}} V_{ik}(\boldsymbol{x}_i^{\tau}, \boldsymbol{x}_k^{\tau}) , \tag{4.16}$$

where the unary term and pairwise terms are

$$D_k(\boldsymbol{x}_k^{\tau}) = \left\| \boldsymbol{F}_p(\boldsymbol{x}_k^{\tau}) - \frac{d^2 \boldsymbol{x}_k^{\tau}}{dt} \right\| - \ln p(\boldsymbol{z}_k^{\tau} | \boldsymbol{x}_k^{\tau}, \Delta \boldsymbol{x}_0^{\tau}) , \tag{4.17}$$

$$V_{ik}(\boldsymbol{x}_i^{\tau}, \boldsymbol{x}_k^{\tau}) = \boldsymbol{F}_r(\boldsymbol{x}_i^{\tau}, \boldsymbol{x}_k^{\tau}) , \tag{4.18}$$

respectively.

4.1.4 Optimization over a Large Number of Pedestrians

In highly congested environments (*e.g.*, $K > 100$), the computational cost for optimizing Eq. (4.7) increases linearly in the number of pedestrians K. To handle realistic scenarios in which most of the pedestrians in the crowd are occluded by others, we use $\tilde{K}$ selected pedestrians whose size is above a predetermined threshold ϵ. We define a set of neighboring pedestrians at time τ $\mathcal{N}(\boldsymbol{x}_0^{\tau}) = \{\boldsymbol{x}_k^{\tau} : \|v_k\| \geq \epsilon\}$.

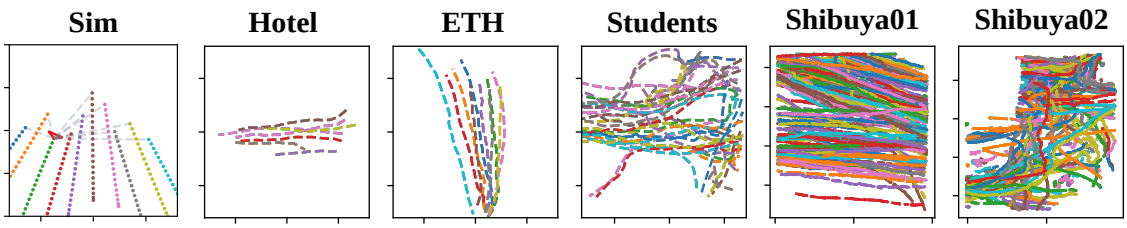


Figure 4.1: **Typical example trajectories.** Typical example trajectories from the datasets Sim, Hotel, ETH, Students, and Shibuya. In the Sim Example, the red triangle is the virtual camera that observes projected pedestrians on the image plane, where dashed gray lines denote the projection.

The energy minimization for the neighboring pedestrians becomes

$$\mathcal{N}(\hat{\boldsymbol{x}}_0^\tau) = \underset{k \in \{k: f_n(u_k^\tau, v_k^\tau) \geq \epsilon\}}{\operatorname{argmin}} \mathcal{E}_p(\boldsymbol{x}_k^\tau; \Delta\hat{\boldsymbol{x}}_0^{\tau_0:\tau}, \boldsymbol{z}_k^\tau, \boldsymbol{x}_k^{\tau_0:\tau-1}) \,. \tag{4.19}$$

Note that optimizing positions of only foreground pedestrians may result in inaccurate localization due to the incomplete interaction model that considers only a small part of the whole crowd. Nevertheless, in Sec. 4.2.3, we show that our proposed framework achieves sufficient localization accuracy even with a small number of selected pedestrians in a super-dense crowd.

4.1.5 Implementation Details

We use the validation split of each crowd dataset [39] to find the optimal hyperparameters of the pedestrian interaction models. We set the weight parameter of the desired force $\boldsymbol{F}_p$ to $\eta = 0.5$, and the variance of the Gaussian potential to $\sigma^2 = 1.0$ for the social force model. For each dataset of simulated and real trajectories, the size of the ground field, where pedestrians are walking from starting points to their destinations, is scaled to $[-8.0, 8.0]$ m. We also assume that the initial positions of pedestrians $\boldsymbol{x}_k^{\tau_1}$ and $\boldsymbol{x}_k^{\tau_1+1}$ for time $\tau_1, \tau_1 + 1$ are given a priori, and the positions at the next timesteps $\mathcal{X}_k^{\tau_1+2:\tau_2} = \{\boldsymbol{x}_k^{\tau_1+2}, \ldots, \boldsymbol{x}_k^{\tau_2}\}$ are sequentially estimated based on our approach.

4.2 Experiments

We validate the effectiveness of the proposed geometric view birdification method through an extensive set of experiments. We constructed several datasets

Table 4.1: **Overview of birdification dataset.** For real trajectories, we selected scenes of Hotel, ETH, and Students by taking into account the number of people in the crowd. "Seq." corresponds to all the frames captured by a moving observer. "Len." denotes the number of frames included in one sequence.

Dataset	Seq.	Len.	People in Crowd			Int.	observer	input	height	occluded
	Total	Avg	Min	Avg	Max	model	view	bboxes	variances	pedestrians
Sim	500	20.0	10	—	50	synthetic	synthetic	given	✔	
Hotel	340	15.0	3	6.31	15	real	synthetic	given	✔	
ETH	346	14.4	3	9.29	26	real	synthetic	given	✔	
Students	849	45.8	13	44.2	75	real	synthetic	given	✔	
Shibuya01	806	317	1	523	770	real	synthetic	given	✔	
Shibuya02	568	299	25	281	492	real	synthetic	given	✔	
GTAV	—	400	3	6	12	synthetic	photorealistic	MOT [90]		✔

consisting of synthetic pedestrian trajectories (**Sim**), real pedestrian trajectories (**Hotel**, **ETH**, **Students**, and **Shibuya**), and photorealistic crowd simulation (**GTAV**). These datasets differ in several aspects (*i.e.*, density of crowd, synthetic view or not, synthetic or real interaction models). Table 4.1 summarizes the statistics and taxonomy of these datasets. We also validate our method on a real mobile robot-view dataset [58] consisting of a pair of real 360° cylindrical images and 2D-3D bounding box annotations of surrounding pedestrians.

4.2.1 View Birdification Datasets

To the best of our knowledge, no public dataset is available for evaluating view birdification (*i.e.*, ego-video in crowds). We construct the following three datasets, which we will publicly disseminate, for evaluating our method and also to serve as a platform for further studies on view birdification.

Synthetic Pedestrian Trajectories The first dataset consists of synthetic trajectories paired with their synthetic projections to an observation camera. This data allows us to evaluate the effectiveness of view birdification when the crowd interaction model is known. The trajectories are generated by the social force model [33] with a varying number of pedestrians ($3 \leq K \leq 15$), and a perspective observation camera mounted on one of them. We set the relaxation parameter η in Eq. (4.13) to be 0.5. To evaluate the validity of our geometric formulation and optimization solution with this dataset, we assume ideal observation of pedestri-

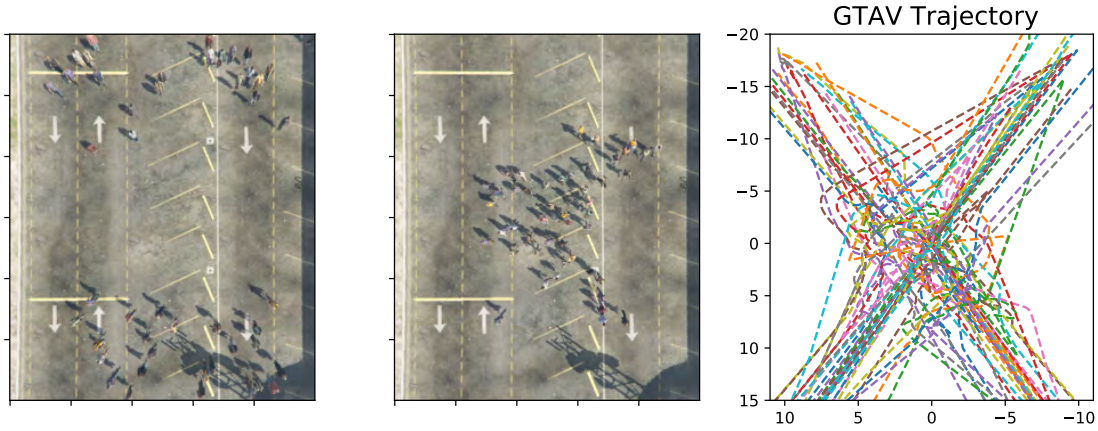


Figure 4.2: **Example Trajectories from the GTAV dataset.** (Left) Pedestrians are spawned at one of the four corners of the field. (Center) Pedestrians walking towards their destinations while avoiding collisions. (Right) Trajectories of each pedestrian in one sequence.

ans, *i.e.*, pedestrians do not occlude each other and their projected heights can be accurately deduced from the observed images. We also assume that the pedestrians are extracted from the ego-centric video perfectly but their heights h_k are sampled from a Gaussian distribution $h_k \sim \mathcal{N}(\mu_h, \sigma_h^2)$ with mean $\mu_h = 1.70$ [m] and a standard deviation $\sigma_h \in [0.00, 0.07]$ [m] based on the statistics of European adults [87].

Real Pedestrian Trajectories The second dataset consists of real pedestrian trajectories paired with their synthetic projections on an observation camera's image-plane. The trajectories are extracted from publicly available crowd datasets: three sets of sequences referred to as **Hotel**, **ETH**, and **Students** are from ETH [73] and UCY [50]. The two referred to as **Shibuya01** and **Shibuya02** are from CroHD dataset [82]. As in the synthetic pedestrian trajectories dataset, we render corresponding ego-centric videos from a randomly selected pedestrian's vantage point. Hotel, ETH, Students, and Shibuya datasets correspond to sparsely, moderately, densely, and super-densely crowded scenarios, respectively. This dataset allows us to evaluate the effectiveness of our method on real data movements.

Photorealistic Crowd Simulation The third dataset consists of synthetic trajectories paired with their photo-realistic projection captured with the limited field of views and frequent occlusions between pedestrians. Evaluation on this dataset

lets us examine the end-to-end effectiveness of our method including robustness to tracking errors. Inspired by previous works on crowd analysis and trajectory prediction [88, 11], we use the video game engine of *Grand Theft Auto V* (GTAV) developed by *Rockstar North* [76] with crowd flows automatically generated from programmed destinations with collision avoidance. We collected pairs of ego-centric videos with 90° field-of-view and corresponding ground truth trajectories on the ground plane using Script Hook V API [80]. We randomly picked 50 different person models with different skin colors, body shapes, and clothes. We prepare two versions of this data, one with manually annotated centerline and heights of the pedestrians in the observed video frames and the other with those automatically extracted with a pedestrian detector [90] pretrained on MOT-16 [62] which includes data captured from a moving platform.

4.2.2 Example Trajectories

Fig. 4.1 visualizes typical example sequences from the synthetic dataset referred to as **Sim** and from the real trajectory dataset referred to as **Hotel**, **ETH**, **Students**, and **Shibuya**. In all of these datasets, a virtual observation camera is assigned to one of the trajectories and the observer captures the rest of the pedestrians in the sequence. Fig. 4.2 shows example trajectories of the GTAV dataset. The size of the ground field, where pedestrians are walking from starting points to their destinations, is configured to be 20m $\times$ 40m. We spawned 50 pedestrians starting from one of the four corners of the field, $[-10, -10]$, $[10, 10]$, $[10, -20]$, $[10, 20]$, and set the opposite side of the field as their destinations. Both the starting points and destinations were randomized with a uniform distribution. In the **GTAV** dataset, an observation camera is mounted on one of the pedestrians walking in the crowd flow and we can obtain pairs of ground-truth trajectories and ego-centric videos with 90° field -of-view via Script Hook V APIs [80].

4.2.3 View Birdification Results

Evaluation Metric. We quantify the accuracy of our method by measuring the differences between the estimated positions of the pedestrians $\boldsymbol{x}_k^\tau$ and the observer $R^\tau, \boldsymbol{x}_0^\tau$ on the ground plane from their ground truth values $\dot{\boldsymbol{x}}_k^\tau, \overline{R}^\tau$, and $\dot{\boldsymbol{x}}_0^\tau$, respectively. The translation error for the observer is $\Delta \boldsymbol{t} = \frac{1}{T}\sum^\tau \|\boldsymbol{x}_0{}^\tau - \dot{\boldsymbol{x}}_0^\tau\|$, where τ is a timestep duration of the sequence. The rotation error of the observer is $\Delta \boldsymbol{r} = \frac{1}{T}\sum_t \arccos(\frac{1}{2}\ \mathrm{trace}(R^\tau(\overline{R}^\tau)^\top - 1)$. We also evaluate the absolute and

Table 4.2: **Birdification results on real trajectories.** Relative and absolute localization errors of pedestrians, $\Delta\tilde{\boldsymbol{x}}, \Delta\boldsymbol{x}$ (top), and camera ego-motion errors, Δr and $\Delta\boldsymbol{t}$ (bottom), were computed for each frame for three different video sequences. Baseline methods only extrapolate movements on the ground plane resulting in missing entries (–). The results demonstrate the effectiveness of our view birdification.

Dataset		**Hotel / sparse**		**ETH / mid**		**Students / dense**	
	σ_h	$\Delta\tilde{\boldsymbol{x}}$ [m]	$\Delta\boldsymbol{x}$ [m]	$\Delta\tilde{\boldsymbol{x}}$ [m]	$\Delta\boldsymbol{x}$ [m]	$\Delta\tilde{\boldsymbol{x}}$ [m]	$\Delta\boldsymbol{x}$ [m]
CV [79]	–	–	0.294 ± 0.186	–	0.275 ± 0.195	–	0.223 ± 0.169
SF [33]	–	–	0.289 ± 0.207	–	0.261 ± 0.174	–	0.222 ± 0.163
VB-CV	0.00	0.051 ± 0.029	0.070 ± 0.030	0.089 ± 0.045	0.115 ± 0.049	0.022 ± 0.008	0.023 ± 0.008
	0.07	0.051 ± 0.029	0.070 ± 0.030	0.090 ± 0.045	0.116 ± 0.050	0.021 ± 0.007	0.022 ± 0.008
VB-SF	0.00	$\mathbf{0.048 \pm 0.027}$	$\mathbf{0.052 \pm 0.033}$	$\mathbf{0.070 \pm 0.040}$	$\mathbf{0.079 \pm 0.047}$	$\mathbf{0.009 \pm 0.003}$	$\mathbf{0.010 \pm 0.006}$
	0.07	$\mathbf{0.049 \pm 0.027}$	$\mathbf{0.052 \pm 0.032}$	$\mathbf{0.071 \pm 0.040}$	$\mathbf{0.080 \pm 0.047}$	$\mathbf{0.009 \pm 0.004}$	$\mathbf{0.010 \pm 0.006}$
	σ_h	Δr [rad]	$\Delta\boldsymbol{t}$ [m]	Δr [rad]	$\Delta\boldsymbol{t}$ [m]	Δr [rad]	$\Delta\boldsymbol{t}$[m]
VB-CV	0.00	$\mathbf{0.015 \pm 0.030}$	0.066 ± 0.089	0.016 ± 0.027	0.095 ± 0.125	$\mathbf{0.001 \pm 0.001}$	0.010 ± 0.007
	0.07	0.017 ± 0.039	0.069 ± 0.100	0.019 ± 0.034	0.110 ± 0.148	$\mathbf{0.001 \pm 0.001}$	0.010 ± 0.007
VB-SF	0.00	0.015 ± 0.036	$\mathbf{0.062 \pm 0.104}$	$\mathbf{0.015 \pm 0.031}$	$\mathbf{0.089 \pm 0.135}$	$\mathbf{0.001 \pm 0.001}$	$\mathbf{0.009 \pm 0.006}$
	0.07	$\mathbf{0.016 \pm 0.042}$	$\mathbf{0.062 \pm 0.103}$	$\mathbf{0.016 \pm 0.035}$	$\mathbf{0.091 \pm 0.153}$	$\mathbf{0.001 \pm 0.001}$	$\mathbf{0.009 \pm 0.006}$

relative reconstruction errors of surrounding pedestrians which are defined by $\Delta\boldsymbol{x} = \frac{1}{K}\frac{1}{T}\sum_k\sum_t \|\boldsymbol{x}_k^\tau - \dot{\boldsymbol{x}}_k^\tau\|$ and $\Delta\tilde{\boldsymbol{x}} = \frac{1}{K}\frac{1}{T}\sum_k\sum_t \|(\boldsymbol{x}_k^\tau - \boldsymbol{x}_0^\tau) - (\dot{\boldsymbol{x}}_k^\tau - \dot{\boldsymbol{x}}_0^\tau)\|$, respectively.

Results on Known Interaction Model. Fig. 4.3 shows the view birdification results on the synthetic trajectories dataset. Although both rotation and translation errors slightly increase as the height standard deviation σ_h becomes larger, the error rate becomes lower as the number of people K increases. This suggests that the more crowded, the more certain the camera position and thus the more accurate the birdification of surrounding pedestrians.

Results on Unknown Real Interaction Models. The real trajectories data allow us to evaluate the accuracy of our method when the interactions between pedestrians are not known. We employ two pedestrian interaction models, Social Force (SF) [33] and ConstVel (CV) [79]. We first evaluate the accuracy of our view birdification (VB) using these models, referred to as *VB-SF* and *VB-CV*, and compare them with baseline prediction models. In these baseline models, referred to as *ConstVel (CV)* and *Social Force (SF)*, we extrapolate a pedestrian position $\mathcal{X}_k^\tau$ from its past locations $\mathcal{X}_k^{\tau-2:\tau-1}$ based on the corresponding interaction model without using the observer's ego-centric view. That is, the baseline model is not view birdification but extrapolation according to pre-defined motion models on

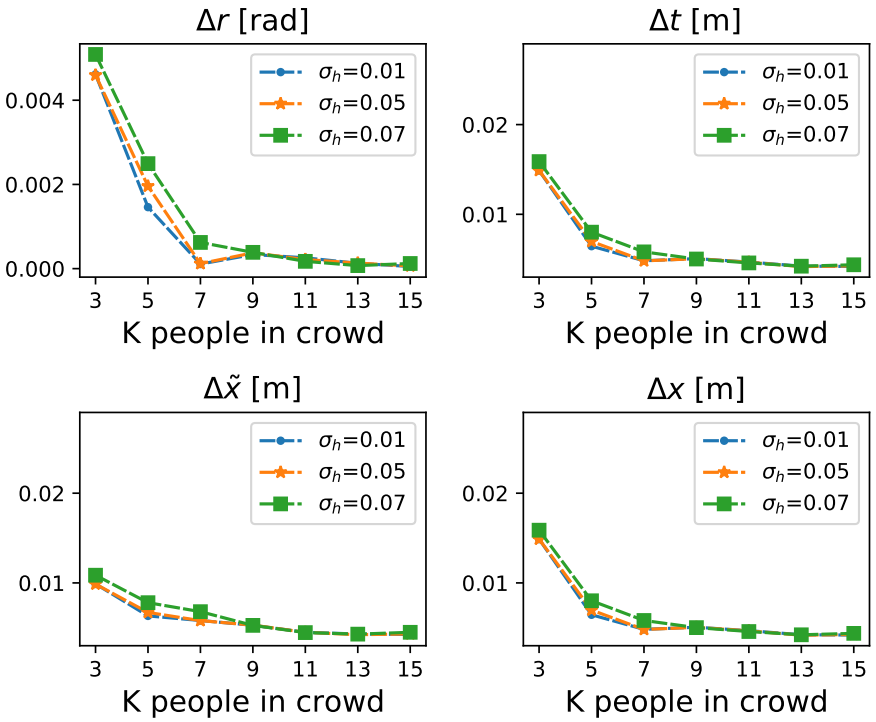


Figure 4.3: **Results on synthetic pedestrian trajectories.** Circle, star, and squared markers denote errors of estimated camera rotations Δr, translations $\Delta \boldsymbol{t}$, relative $\Delta \tilde{\boldsymbol{x}}$ and absolute localization errors $\Delta \boldsymbol{x}$, respectively, with standard deviations of pedestrian heights, $\sigma = 0.01, 0.05, 0.07$ [m], respectively.

the ground plane.

Table 4.2 shows the errors of our method and baseline models. These results clearly show that our method, both *VB-CV* and *VB-SF*, can estimate the camera ego-motion and localize surrounding people more accurately, which demonstrates the effectiveness of birdifying the view and exploiting the geometric constraints on the pedestrians through it. *VB-SF* performs better than *VB-CV* especially in scenes with rich interactions such as ETH and Students, while they show similar performance on the Hotel dataset that includes fewer interactions. Both *VB-SF* and *VB-CV* show accurate camera ego-motion results in the Students dataset, which demonstrates the robustness of ego-centric view localization regardless of the assumed pedestrian interaction models. Our method achieves high accuracy on all three datasets across different standard deviations of heights $\sigma_h \in [0.00, 0.07]$. This also shows that the method is robust to variation in human heights.

Table 4.3: **Birdification results in the super-dense crowd.** Relative and absolute localization errors of pedestrians, $\Delta\tilde{\boldsymbol{x}}, \Delta\boldsymbol{x}$ (top), and camera ego-motion errors, Δr and $\Delta\boldsymbol{t}$ (bottom), were computed for each frame for three different video sequences. Baseline methods only extrapolate movements on the ground plane resulting in missing entries (–). The results demonstrate the effectiveness of our view birdification even in super-dense crowds.

Dataset		**Shibuya01**		**Shibura02**	
	σ_h	$\Delta\tilde{\boldsymbol{x}}$ [m]	$\Delta\boldsymbol{x}$ [m]	$\Delta\tilde{\boldsymbol{x}}$ [m]	$\Delta\boldsymbol{x}$ [m]
CV [79]	–	–	0.221	–	0.245
SF [33]	–	–	0.220	–	0.249
VB-CV	0.07	0.023	0.024	0.025	0.026
VB-SF	0.07	0.022	0.023	0.024	0.025
	σ_h	$\Delta\boldsymbol{r}$ [rad]	$\Delta\boldsymbol{t}$ [m]	$\Delta\boldsymbol{r}$ [rad]	$\Delta\boldsymbol{t}$ [m]
VB-CV	0.07	0.001	0.011	0.001	0.012
VB-SF	0.07	0.001	0.011	0.001	0.011

Selecting pedestrians in Super Dense Crowds. Table 4.3 shows the localization errors of the view birdification and the baseline models on the super-dense crowd datasets (Shibuya01, Shibuya02). In a highly congested scenario $K > 100$, we can no longer consider all the pedestrians due to computational cost. Following Sec.4.1.4, for each frame, we select a set of pedestrians in the neighborhood of the camera $\mathcal{N}(\boldsymbol{x}_0)$ with a threshold $\|v_k\| \geq \epsilon$ and $\epsilon = 5.0$. Even if we do not consider all the pedestrians in the crowd for optimizing with the assumed interaction model, our model *VB-CV* and *VB-SF* still demonstrate comparable localization accuracy with those for the Students dataset. This is because our assumed interaction model considers up to first-order interactions with other pedestrians in close proximity, and the interaction models with selected neighboring pedestrians can approximate the whole interaction models in super dense crowd with sufficient accuracy. These results clearly demonstrate our model can be applied to highly dynamic crowded environments, where static keypoints-based SLAM fails.

Photorealistic Crowds. Fig. 4.4 shows qualitative results on the photorealistic crowd dataset. Considering more practical use cases, we evaluate the accuracy of our method in the existence of detection noises. We prepared two versions of inputs, one manually annotated with centerlines of the people and their heights and

Table 4.4: **Quantitative Results on GTAV dataset for different inputs.** The relative and absolute localization errors of pedestrians, $\Delta\tilde{\boldsymbol{x}}$ and $\Delta\boldsymbol{x}$, respectively, and the errors of camera ego-motion estimation, Δr, and $\Delta\boldsymbol{t}$, computed for each frame whose mean values are shown. **cline** denotes ideal, manually annotated inputs and **MOT** denotes inputs with detection noise by multi-object tracker [90].

Input	$\Delta\boldsymbol{r}$ [rad]	$\Delta\boldsymbol{t}$ [m]	$\Delta\tilde{\boldsymbol{x}}$ [m]	$\Delta\boldsymbol{x}$ [m]
cline(manual)	0.015	0.097	0.441	0.491
MOT [90]	0.016	0.101	0.491	0.530

the other with those automatically extracted from the off-the-shelf multi-object tracker [90]. We prepared two versions of inputs, one manually annotated with centerlines of the people and their heights and the other with those automatically extracted from a multi-object tracker (MOT). We compared view birdification results using these two different inputs, which are referred to as *VB-cLine* and *VB-MOT*. As shown in the top two rows, *VB-MOT* accurately estimates camera ego-motion and on-ground positions of automatically detected pedestrians with an off-the-shelf tracker. People tracked in more than three frames are birdified. Even with occlusions in the image and noisy height estimates computed from detected bounding boxes, our approach robustly estimates the camera ego-motion and surrounding pedestrian positions. Due to perspective projection, localization error caused by erroneous detection in the image plane is proportional to the ground-plane distance between the camera and the detected pedestrian. We further compared these results with *VB-cline* as shown in the bottom two rows Fig. 4.4 to highlight the effect of automatically detecting the pedestrians for view birdification (*i.e.,* to see how the results change if the pedestrian heights were accurate). The resulting accuracies are comparable, which demonstrates the end-to-end effectiveness. To further ameliorate the errors caused by detection noises, our method can also be extended, for instance, by replacing the noise model in Eq. (4.3) with a 2D Gaussian distribution.

4.2.4 Unknown Ego-Motion Recovery with the Real Mobile Platform Dataset

JackRabbot Dataset. We also test on the JackRabbot Dataset and Benchmarks (JRDB) [58], which includes panorama (360°) RGB images with 2D-3D bounding

box annotations of pedestrians captured by a mobile robot platform of human-compatible size. The robot captures the social interactions of a crowd in outdoor/indoor environments, where all the pedestrian IDs are assigned and their 3D locations are annotated in the relative coordinate system of the mounted camera. The camera ego-motion is constrained to the 2D motion on the ground, *i.e.*, $R \in SO(2)$, $\boldsymbol{t} \in \mathbb{R}^2$. The notable difference from our view birdification datasets is, the motion model of the ego-motion does not conform with the crowd motion model of surrounding pedestrians. This dataset allows us to evaluate the applicability of our method on mobile robot platforms with unknown motion model. In this dataset, we use the cylindrical projection model described in Eq. (3.3) for 360° cylindrical RGB image inputs, and reconstruct both the ego-motion and pedestrian trajectories in absolute coordinates only from observed 2D movements in the image.

Comparison with the robot localization results from sensor values. We compare the localization results with that estimated from IMU sensor values and the wheel odometry recorded in the rosbag of the dataset. As no ground-truth ego-motion is available for this dataset, we create pseudo localization results by fusing these sensor values with an extended Kalman Filter [64]. Fig. 4.5 demonstrates our view birdification results with an unknown ego-motion model. Our method can successfully recover the on-ground absolute trajectories of both the camera and its surrounding pedestrians. Even in these scenarios in which the camera ego-motion model is not consistent with the assumed crowd motion model (*e.g.*, a mobile robot platform), our method can recover the camera ego-motion as long as the camera observes the pedestrians with an assumed motion model. Both sensor-based and our vision-based localization have uncertainties arising from the observation errors, which often results in a significant ego-motion drift in long-term navigation. Even if the mobile platform is equipped with an IMU and other odometry sensors, the birdification results are still essential for obtaining reliable ego-motion estimates and can provide a reliable source for sensor fusion. The quantitative gap between our estimated ego-motion and IMU values processed with Kalman Filter was $\Delta r = 0.001$ [rad] and $\Delta \boldsymbol{t} = 0.023$ [m] on average in a tested sequence.

Comparison with learning-based monocular depth estimation. We compared

the accuracy of pedestrian localization by our method with those estimated by the state-of-the-art monocular depth estimator [8] which uses inverse projection constraints similar to our method. The monocular depth estimator is pretrained on KITTI dataset and takes pedestrian keypoints as inputs calculated by an external keypoint detector [46]. For fair comparison, we apply our method to pedestrians detected by the same keypoint-based detector. In the JRDB sequence used in the previous paragraph, the accuracy of pedestrian localization by the learning-based estimator [8] is $\Delta \boldsymbol{x} = 1.789 \pm 1.540$ [m], while that by our method is $\Delta \boldsymbol{x} = 0.482 \pm 0.350$ [m] on average. The learning-based estimator shows poor accuracy compared to our method, and the variance of the localization accuracy is an order of magnitude larger than that of our method. While our proposed method sequentially estimates the location of each pedestrian taking their motion model into account, the monocular depth estimator does not consider temporal consistency, which results in higher variance of localization accuracy.

4.3 Discussion

In this chapter, we propose a novel method for on-ground trajectory reconstruction of both the camera and pedestrians only from perceived movements of dynamic objects, *i.e.,* pedestrians. This allows us to recover the camera ego-motion even in a dense crowd, where static keypoints are occluded by surrounding pedestrians. One may think "even if the static keypoints near the camera are occluded and unable to track, we can still track backgrounds far from the crowd, *i.e.,* buildings". Fig. 4.7 shows typical example cases in the GTAV dataset. The static backgrounds (*i.e.,* buildings and trees) are detected and tracked, but are located far away (≥ 30 [m]) from the observation camera.

We simulate the robustness of a geometric relative pose solver [71] against noise according to the distances between the keypoints and the observation camera, and compare the accuracy with our approach based on dynamic keypoints. We generate 100 static keypoints with uniform distribution in a voxel grid $\boldsymbol{V}_{w,h,d}$, where width $w = 20$ [m], height $h = 10$ [m], and depth $d = 5$ [m]. The keypoints are captured by two cameras located at $[2 : 40]$ [m], where these two camera poses are randomly generated with conditions $\Delta r \leq 0.20$ [rad] and $\Delta \boldsymbol{t} \leq 1.0$ [m]. To test the robustness against detection noises in the image, we add uniform random noise ranging between [-1:1] [px] at each pixel and apply the five-point relative pose solver [71] with RANSAC [23]. At every distance from the camera, we test

100 trials of relative pose solver for static keypoints randomly generated at each trial.

Fig. 4.6 shows the errors of relative pose estimation for rotation Δr and translation $\Delta \boldsymbol{t}$. These results clearly show that the further the keypoints, the worse the accuracy of pose estimation. Although the relative pose solver performs well when static keypoints are observed at nearly ≤ 2 [m], the translation errors are as worse as 0.50 [m] on average when at ≥ 6 [m]. We also compare these results with our view birdification results including detection noises by the external multi-object tracker [90]. The red dotted line indicates the accuracy of our view birdification with multi-object tracking noises on the GTAV dataset (Table 4.4). Even with the detection noises, the errors of rotation and translation are 0.016 [rad] and 0.097 [m], respectively, which are significantly lower than those obtained with a relative pose solver with static keypoints observed from far away (≥ 6 [m]).

These results clearly indicate that *nearby dynamic keypoints are better than distant static keypoints* for camera pose estimation in densely crowded environments. More specifically, if observed static keypoints are at as far as ≥ 6 [m], our proposed view birdification based on the movements of nearby pedestrians performs better.

4.4 Failure Cases

We also analyze failure cases of our view birdification to understand the limitations of the method. For this, we picked sequences from **ETH** data that showed a high error rate in terms of camera localization. Fig. 4.8 visualizes posterior distributions of the observer location $p(\boldsymbol{x}_0^\tau | \mathcal{Z}_{1:K}^\tau, \mathcal{X}_{0:K}^{\tau-1})$ and surrounding pedestrians $\int_{\boldsymbol{x}_0^\tau \in \mathcal{X}_s} p(\mathcal{X}_{1:K}^\tau | \mathcal{Z}_{1:K}^\tau, \boldsymbol{x}_0^\tau) p(\boldsymbol{x}_0^\tau) d\boldsymbol{x}_0^\tau$ by sampling $\boldsymbol{x}_0^\tau \in \mathcal{X}_s$ in Eq. (4.3) and Eq. (4.4), respectively. The first and third rows depict the ground truth trajectories of the camera and pedestrians from τ to $\tau + 9$. The number of pedestrians changes from $K = 3$ to $K = 5$. The second and fourth rows visualize the posterior distributions for each of those two rows. As can be observed in the posteriors shown in the second row, the estimated observer location becomes a heavy-tailed distribution when the number of pedestrians in the crowd is small ($K = 3$). In contrast, as shown in the fourth row, the posterior distribution becomes sharper when the crowd is denser ($K = 5$). The ambiguity of localization increases when pedestrians walk almost parallel to the observer (*e.g.*, timesteps $\tau = \tau + 2$ and $\tau + 3$). In contrast, the posterior distribution becomes sharp again when the camera ob-

serves more pedestrians walking in diverse directions.

Figure 4.4: **Results on photorealistic crowd dataset.** The top row shows detected pedestrians with a multi-object tracker in bounding boxes and the third row shows manually annotated human heights (center lines). The figures in the second and fourth rows depict view birdification results for them. Colors correspond to Pedestrian IDs. Red triangles denote camera position estimates $\boldsymbol{x}_0^\tau$ and dashed circles denote estimated pedestrian positions $\boldsymbol{x}_k^\tau$ at time τ. Grey triangles and circles denote ground-truth camera and pedestrian positions, respectively. View birdificaiton results for both automatic and manually detected people show consistently high accuracy. These results demonstrate the end-to-end accuracy of view birdification.

Figure 4.5: **Results on real robot dataset.** The top four rows show 2D bounding box annotations for pedestrians in the cylindrical RGB image at $\tau \in (\tau, \tau + 10, \tau + 20, \tau + 30, \tau + 40)$. The fourth row depicts ground-truth global layout of pedestrians relative to the camera $\boldsymbol{x}_0 = [0, 0]^\top$ at every timestep. The fifth row shows the view birdification results given the sequence of 2D bounding box movements in the cylindrical RGB images. Colors correspond to pedestrian IDs. Red triangles denote camera position estimates.

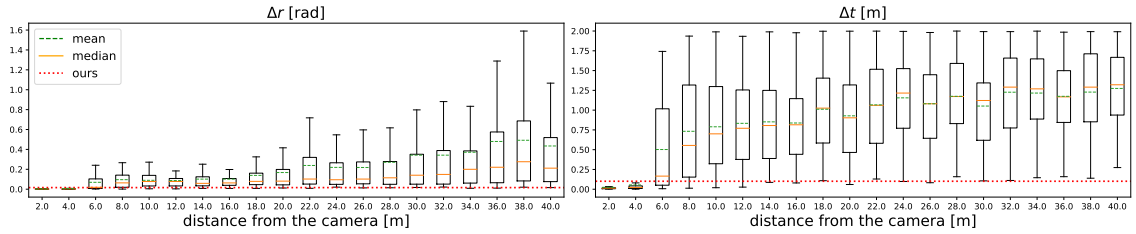


Figure 4.6: **Robustness of relative pose solver [71] against noise according to the distances from keypoints.** These two figures show errors in pose estimation consisting of rotation Δr [rad] and translation $\Delta \boldsymbol{t}$ [m]. The boxes indicate the range between 25th and 75th percentiles from the lowest values, where the orange and green dashed lines medians and means of the errors, respectively. The black whiskers extends from the lowest to highest values. The red dotted line indicates the error of our method with MOT input noise (Table 4.4). While the geometric solver works well with keypoints captured at ≤ 2 [m], the accuracy of relative pose estimation significantly drops when captured at ≥ 6 [m].

Figure 4.7: **Static keypoints trackable in crowded scenes.** In these typical two cases, static backgrounds far from the crowd are detected and tracked, while keypoints near the camera are untrackable due to severe occlusions. These static keypoints are at 30[m] from the observation camera.

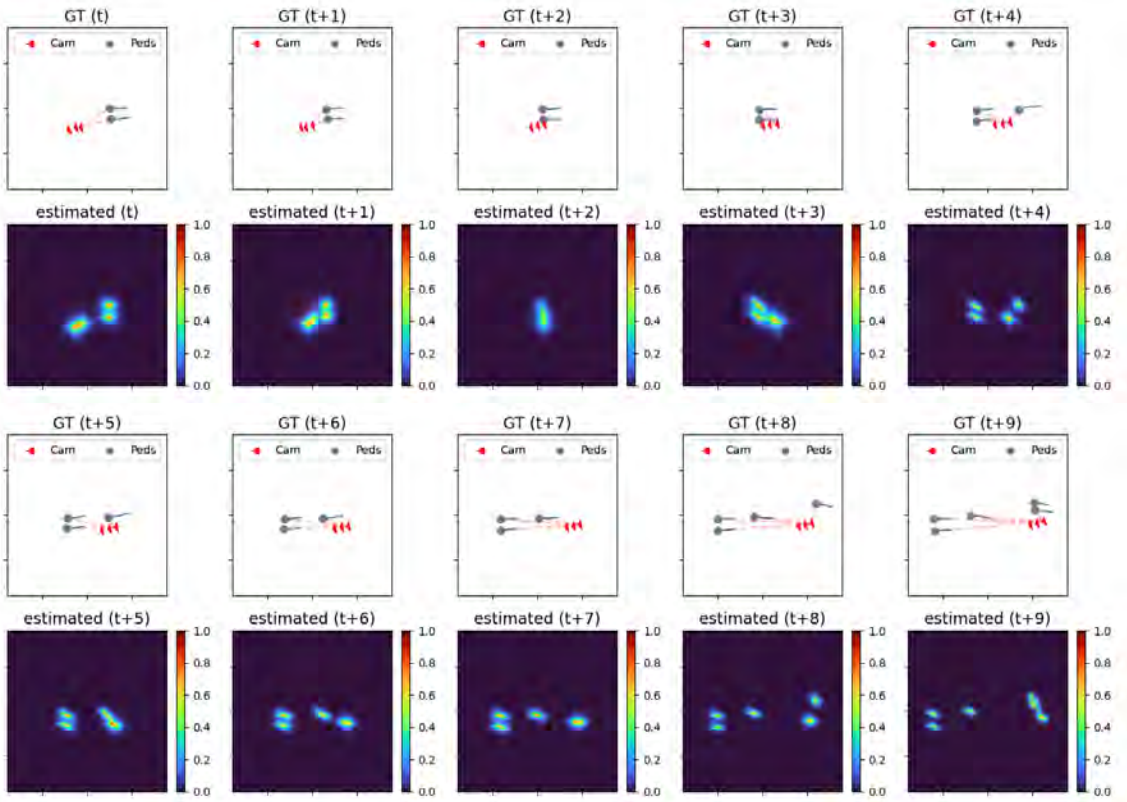


Figure 4.8: **Visualization of posterior distributions of the ETH dataset.** (First and third rows) Ground truth trajectories of the camera and its surrounding pedestrians. (Second and fourth rows) Visualization of posterior distributions of the location of the observer $\boldsymbol{x}_0^\tau$ and surrounding pedestrians $\boldsymbol{x}_k^\tau$. The heatmaps correspond to low (blue) to high (red) probabilities.

Chapter 5

Learning to recover ground-plane crowd trajectories and ego-motion

5.1 Background

We as human beings have a fairly accurate idea of the absolute movements of our surroundings in the world coordinate frame, even when we can only observe their movements relative to our own in our sight such as when walking in a crowd. Enabling a mobile agent to maintain a dynamically updated map of surrounding absolute movements on the ground, solely from observations collected from its own vantage point, would be of significant use for various applications including robot navigation [69], autonomous driving [49], sports analysis [17], and crowd monitoring [26, 39, 60]. The key challenge lies in the fact that when the observer (*e.g.*, person or robot) is surrounded by other dynamic agents, static "background" can hardly be found in the agent's field of view. In such scenes, conventional visual localization methods including SLAM would fail since static landmarks become untrackable or unreliable due to frequent occlusions and observation noises sensitive to distances [67]. External odometry signals such as IMU and GPS are also often unreliable. Even when they are available, visual feedback becomes essential for robust pose estimation (imagine walking in a crowd with closed eyes).

In Chapter 4, we introduced this exact task as *view birdification* whose goal is to recover on-ground trajectories of a camera and a crowd just from perceived movements (not appearance) in an ego-centric video [66] [1]. They proposed to

[1]Note that Bird's Eye View transform is a completely different problem as it concerns a single frame view of the appearance (not the movements) and cannot reconstruct the camera ego-motion.

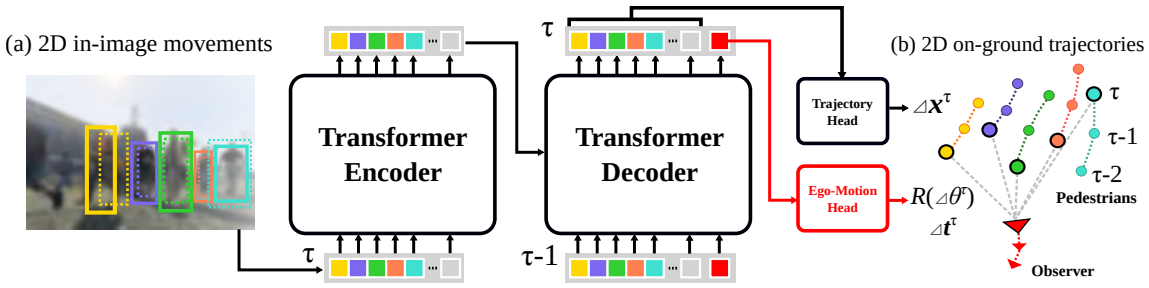


Figure 5.1: Given bounding boxes of moving pedestrians in an ego-centric view captured in the crowd, ViewBirdiformer reconstructs on-ground trajectories of both the observer and the surrounding pedestrians. 2023 ©IEEE [68]

decompose these two types of trajectories, one of the pedestrians in the crowd and another of a person or mobile robot with an ego-view camera, with a cascaded optimization which alternates between estimating the displacements of the camera and estimating those of surrounding pedestrians while constraining the crowd trajectories with a pre-determined crowd motion model [33, 79]. This iterative approach suffers from two critical problems which hinder their practical use. First, its iterative optimization incurs a large computational cost which precludes real-time use. Second, the analytical crowd model as a prior is restricting and not applicable to diverse scenes where the crowd motion model is unknown.

In this chapter, we propose *ViewBirdiformer*, a Transformer-based view birdification method. Instead of relying on restrictive assumptions on the motion of surrounding people and costly alternating optimization, we define a Transformer-based network that learns to reconstruct on-ground trajectories of the surrounding pedestrians and the camera from a single ego-centric video while simultaneously learning their motion models. As fig. 5.1 depicts, ViewBirdiformer takes in-image 2D pedestrian movements as inputs, and outputs 2D pedestrian trajectories and the observer's ego-motion on the ground plane. The multi-head self-attention on the motion feature embeddings of each pedestrian of ViewBirdiformer captures the local and global interactions of pedestrians. At the same time, it learns to reconstruct on-ground trajectories from observed 2D motion in the image with cross-attention on features coming from different viewpoints.

A key challenge of this data-driven view birdification lies in the inconsistency of coordinate frames between input and output movements—the input is 2D in-image movements of pedestrians relative to ego-motion, but the expected outputs are on-ground trajectories in absolute coordinates (*i.e.*, independent of the

observer's motion). ViewBirdiformer resolves this by introducing the two types of queries, *i.e.,* the camera ego-motion and pedestrian trajectories, in a multi-task learning formulation, and by transforming coordinates of pedestrian queries relative to the previous ego-motion estimates.

We thoroughly evaluate the effectiveness of our method using the view birdification dataset [66] and also by conducting ablation studies which validate its key components. The proposed Transformer-based architecture learns to reconstruct trajectories of the camera and the crowd while learning their motion models by adaptively attending to movement features of them in the image plane and on the ground. It enables real-time view birdification of arbitrary ego-view crowd sequences in a single inference pass, which leads to three orders of magnitude speedup from the iterative optimization approach [66]. We show that the results of ViewBirdiformer can be opportunistically refined with geometric post-processing, which results in similar or better accuracy than state-of-the-art [66] but still in orders of magnitude faster execution time.

5.2 Preliminary

Let us recall the definition of view birdification. We have a crowd of people and one observer in the crowd with an ego-centric camera observing the surroundings while moving around. The observer can either be one of the pedestrians of the crowd or a mobile robot, or even an autonomous vehicle, in the crowd. As the observer is immersed in the crowd with a limited but dynamic field-of-view, the static background cannot be reliably found in the ego-centric view.

Let us assume that the crowd consists of N people. We set the z-axis of the world coordinate system to the normal of the ground plane (xy-plane). As in previous work [66], we assume the ground plane to be planar and the observer's camera direction is parallel to it. We can assume this without loss of generality as the camera pitch and roll can be corrected either by measurements of the moment (*e.g.,* with an IMU) [7]. View birdification thus is the problem of recovering 2D trajectories of the observer and surrounding people (visible in the ego-centric view) on the ground plane (xy-plane) from their 2D in-image movements in the ego-centric view.

Let $\boldsymbol{x}_i = [x_i, y_i]^\top$ denote the on-ground location of the i^{th} pedestrian and $\boldsymbol{\pi} = \left[c_x, c_y, \theta_z\right]$ the pose of the observer's camera. The ego-centric camera pose $\boldsymbol{\pi}$ consists of a rotation matrix $R(\theta_z) \in \mathbb{R}^{2\times 2}$ parameterized by the rotation angle

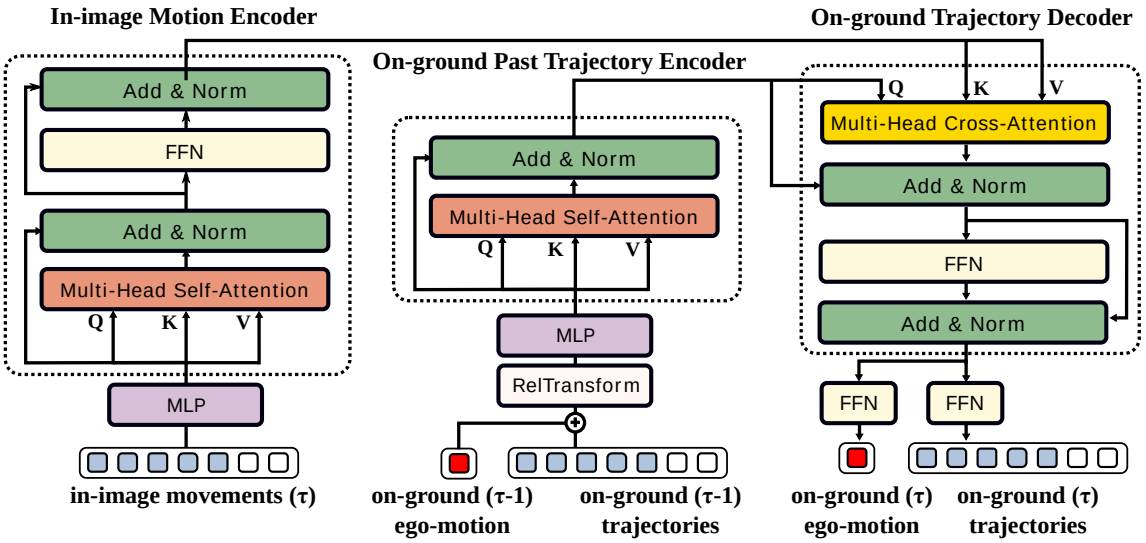


Figure 5.2: **The overall architecture of ViewBirdiformer.** The decoder takes two types of queries: camera queries and pedestrian queries. These queries are fed autoregressively from the previous frame output embeddings of the last decoding layer. 2023 ©IEEE [68]

around the z-axis θ_z and 2D translation $\boldsymbol{t} = -R(\theta_z)\left[c_x, c_y\right]^\top$, *i.e.*, the viewing direction and camera location on the ground, respectively. The observer's camera location is $\left[c_x, c_y, c_z\right]^\top$, where the mounted height c_z is constant across frames, and the intrinsic matrix $A \in \mathbb{R}^{3\times3}$ is assumed to be constant.

At every timestep τ, we extract the state of each pedestrian $\boldsymbol{s}_i^\tau$ for all those visible in the observed image, $n \in \{1, 2, \ldots, N\}$. The pedestrian state encodes the two-dimensional center of the pedestrian's bounding box and the velocity calculated by its displacement from the bounding box center of the previous $(\tau - 1)$ frame. These states of visible pedestrians in the ego-centric view $\mathcal{S}_i^{\tau_1:\tau_2}$ can be extracted with an off-the-shelf multi-object tracker with consistent IDs. Given a sequence of in-image pedestrian states $\mathcal{S}_i^{\tau_1:\tau_2} = \{\boldsymbol{s}_i^{\tau_1}, \boldsymbol{s}_i^{\tau_1+1}, \ldots, \boldsymbol{s}_i^{\tau_2}\}$ from timestep τ_1 to τ_2, our goal is to simultaneously reconstruct the on-ground trajectories of pedestrians $\mathcal{X}_i^{\tau_1:\tau_2} = \{\boldsymbol{x}_i^{\tau_1}, \boldsymbol{x}_i^{\tau_1+1}, \ldots, \boldsymbol{x}_i^{\tau_2}\}$ and the observer's camera poses $\Pi^{\tau_1:\tau_2} = \{\boldsymbol{\pi}^{\tau_1}, \boldsymbol{\pi}^{\tau_1+1}, \ldots, \boldsymbol{\pi}^{\tau_2}\}$.

5.3 ViewBirdiformer

Our goal is to devise a method that jointly transforms the 2D in-image movements into 2D on-ground trajectories and models the on-ground interactions between

pedestrians in a single framework. For this, we formulate view birdification as a set-to-set translation task, and derive a novel Transformer-based network referred to as *ViewBirdiformer.*

5.3.1 Geometric 2D-to-2D Transformer

Given a sequence of in-image pedestrian states for N people in a crowd at time τ, we first embed them into a set of d-dimensional state feature vectors $\mathcal{F}_s \in \mathbb{R}^{N\times d}$ with a multilayer perceptron (MLP). We similarly embed past $(\tau-1)$ on-ground trajectories of the pedestrians and the observer's camera, too. ViewBirdiformer consists of an encoder that encodes input in-image state features $\mathcal{F}_s$ into a sequence of hidden state features $\mathcal{H}_s \in \mathbb{R}^{N\times d}$, and a decoder that takes in the hidden features and on-ground queries $\mathcal{Q}_o \in \mathbb{R}^{(N+1)\times d}$

$$\mathcal{H}_s = \mathcal{E}_\psi(\mathcal{F}_s), \quad \mathcal{F}_o = \mathcal{D}_\phi(\mathcal{Q}_o, \mathcal{H}_s)\,, \tag{5.1}$$

where $\mathcal{E}_\psi$ and $\mathcal{D}_\phi$ are the encoder and decoder models with learnable model parameters ψ and ϕ, respectively. Figure 5.2 depicts the overall architecture of our ViewBirdiformer.

Attention layers A standard attention mechanism [85] accepts three types of inputs: a set of queries $\mathcal{Q} \in \mathbb{R}^{M\times d}$, a set of key vectors $\mathcal{K} \in \mathbb{R}^{N\times d}$, and a set of value embeddings $\mathcal{V} \in \mathbb{R}^{N\times d}$. The output is computed by values weighted by an attention matrix $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ composed of dot-products of queries and keys, and we use softmax to normalize the attention weights,

$$\text{Attn}(\mathcal{Q},\mathcal{K},\mathcal{V}) = \sum_{j=1}^{N} A_{ij}\boldsymbol{v}_j\,,\ \boldsymbol{A}_{ij} = \frac{\exp(\boldsymbol{q}_i^\top \boldsymbol{k}_j)}{\sum_{j'=1}^{N}\exp(\boldsymbol{q}_i^\top \boldsymbol{k}_{j'})}\,. \tag{5.2}$$

The query $\boldsymbol{q}$, key $\boldsymbol{k}$, and value $\boldsymbol{v}$ vectors are linear embeddings of the source $\boldsymbol{f}_s$ and target $\boldsymbol{f}_t$ input state features

$$\boldsymbol{q} = W_q(\boldsymbol{f}_t)\,, \quad \boldsymbol{k} = W_k(\boldsymbol{f}_s)\,, \quad \boldsymbol{v} = W_v(\boldsymbol{f}_s)\,, \tag{5.3}$$

where W_q, W_k, and W_v are linear embedding matrices specific to the vector types. We refer to the case of $\boldsymbol{f}_s = \boldsymbol{f}_t$ as *self-attention*, and the other case $\boldsymbol{f}_s \neq \boldsymbol{f}_t$ as *cross-attention*.

Attention Mask To handle the varying number of pedestrians entering and leaving the observer's view, we apply a mask $\boldsymbol{M}^\tau \in \mathbb{R}^{M\times N}$ to the attention matrix $\boldsymbol{A}^\tau$ as $M^\tau \odot \boldsymbol{A}^\tau$, where $\odot$ denotes Hadamard product. The element of the mask M_{ij}^τ is set to **0** if either of the pedestrians i or j are missing at time τ, otherwise **1**. This allows us to handle temporarily occluded pedestrians.

In-Image Motion Encoder The encoder architecture consists of a single multi-head self-attention layer [85] and a feed-forward network (FFN) layer. We define the input pedestrian states as $\boldsymbol{s}_i^\tau = [p_x, p_y, \Delta p_x, \Delta p_y]^\top$, consisting of the 2D center of the detected bounding box $\boldsymbol{p} = [p_x, p_y]^\top$ and is its velocity $\Delta\boldsymbol{p} = [\Delta p_x, \Delta p_y]^\top$. The encoder $\mathcal{E}_\psi$ computes self-attention over all queries generated by input state feature embeddings $\mathcal{F}_s$, which encodes the interactions between observed pedestrians in image space.

On-ground Trajectory Decoder The Transformer decoder $\mathcal{D}_\phi$ integrates the self-attention based on-ground motion model and the cross-attention between on-ground and ego-views. First, the **On-Ground Past Trajectory Encoder** applies self-attention over queries $\mathcal{Q}$ consisting of an ego-motion query $\boldsymbol{q}_\pi^{\tau-1}$ and on-ground pedestrian trajectory queries $\{\boldsymbol{q}_1^{\tau-1}, \ldots, \boldsymbol{q}_N^{\tau-1}\}$ extracted from previous estimates at $\tau - 1$. We calculate these with on-ground queries $\boldsymbol{q}_\pi^{\tau-1} = W_q(\mathrm{MLP}(\Delta\boldsymbol{\pi}^{\tau-1}))$ and $\boldsymbol{q}_n^{\tau-1} = W_q(\mathrm{MLP}(\boldsymbol{x}_i^{\tau-1} \oplus \Delta\boldsymbol{x}_i^{\tau-1}))$, respectively. The attention learns to capture the implicit local and global interactions of all pedestrians to better predict the future location from past trajectories. Second, the cross-attention layer accepts hidden state features $\mathcal{H}_s$ processed by the encoder and on-ground trajectory queries $\mathcal{Q}_o$ processed by the self-attention layer. This layer outputs feature embeddings $\mathcal{F}_o \in \mathbb{R}^{(N+1)\times d}$ by incorporating features from the ego-centric view. The output $\mathcal{F}_o$ is decoded to the camera ego-motion Π^τ and N pedestrian trajectories $\{\mathcal{X}_1^\tau, \ldots, \mathcal{X}_N^\tau\}$ by task-specific heads. The trajectory decoder is autoregressive, which outputs trajectory estimates one step at a time and feeds the current estimates back into the model as queries to produce the trajectories of the next timestep.

5.3.2 Relative Position Transformation

A key challenge of view birdification lies in the inconsistency of coordinate systems between input and output trajectories. Unlike conventional frame-by-frame 2D-to-3D lifting [8] or image-based bird's eye-view transformation [96], once

the viewpoint of the observer's camera is changed, the observed movements of pedestrians in the image change dramatically. To encourage the network to generalize over diverse combinations of trajectories and observer positions, we transform all the on-ground pedestrian queries relative to the previous $\tau - 1$ observer's camera estimates at every timestep,

$$\tilde{\boldsymbol{x}}_i^\tau = R(\theta_z^{\tau-1})\boldsymbol{x}_i^\tau + \boldsymbol{t}^{\tau-1}, \tag{5.4}$$

$$\Delta\tilde{\boldsymbol{x}}_i^\tau = R(\theta_z^{\tau-1})(\boldsymbol{x}_i^\tau - \boldsymbol{x}_i^{\tau-1}), \tag{5.5}$$

where $\boldsymbol{t}^{\tau-1} = -R(\theta_z^{\tau-1})[c_x^{\tau-1}, c_y^{\tau-1}]^\top$ is the camera translation. We force all on-ground trajectory coordinates to be centered on the observer's camera by defining positions and velocities relative to the observer's camera $\tilde{\boldsymbol{x}} \oplus \Delta\tilde{\boldsymbol{x}}$ as pedestrian features, and the camera displacements $\Delta\boldsymbol{\pi} = [\Delta c_x, \Delta c_y, \Delta\theta_z]^\top$ as the observer's feature.

5.3.3 Ego-motion Estimation by Task-specific Heads

To achieve simultaneous recovery of pedestrian trajectories and ego-motion of the observer's camera, we formulate birdification as a multi-task learning problem. Given a set of past queries $\{\boldsymbol{q}_c^{\tau-1}, \boldsymbol{q}_1^{\tau-1}, \ldots, \boldsymbol{q}_N^\tau\}$ consisting of trajectories of the observer and surrounding pedestrians, the decoder transforms the joint set of camera and pedestrian queries into output embeddings $\mathcal{F}_o \in \mathbb{R}^{(N+1)\times d}$. The output embeddings $\mathcal{F}_o$ consist of two types of features: (i) ego-motion embedding $\mathcal{F}_{\text{ego}} \in \mathbb{R}^{1\times d}$ from which the motion of the observer's camera on the ground is recovered, and (ii) pedestrian trajectory embeddings $\mathcal{F}_{\text{traj}} \in \mathbb{R}^{N\times d}$ represented in a relative coordinate system, where the origin is the position of the camera. These two queries calculated from the previous $t - 1$ frame are decoded simultaneously. We define individual loss functions for these two tasks.

Ego-Motion Loss The ego-motion output embedding $\mathcal{F}_{\text{ego}}$ is decoded into $\Delta\boldsymbol{\pi} = [\Delta c_x, \Delta c_y, \Delta\theta_z]^\top$ by a single feed-forward network. For a batch $\{\Delta\boldsymbol{\pi}^\tau, \ldots, \Delta\boldsymbol{\pi}^T\} \in \mathbb{R}^{3\times T}$ of duration T, we compute the mean squared error

$$\mathcal{L}_{\text{ego}} = \sum_{\tau=1}^{T} \|\Delta\dot{\boldsymbol{\pi}}^\tau - \Delta\boldsymbol{\pi}^\tau\|, \tag{5.6}$$

where $\dot{\boldsymbol{\pi}}$ is the ground-truth camera pose of an observer.

Pedestrian Trajectory Loss Pedestrian trajectory embeddings $\mathcal{F}_{\text{traj}}$ are decoded into 2D positions and velocities $\tilde{\boldsymbol{x}} \oplus \Delta\tilde{\boldsymbol{x}} \in \mathbb{R}^4$ relative to the observer's camera. Given a batch of N observed pedestrians for duration T, we define the trajectory loss function as

$$\begin{aligned}\mathcal{L}_{\text{traj}} = \sum_{\tau=1}^{T}\sum_{i=1}^{N} &\|\dot{\boldsymbol{x}}_i^\tau - (R(\theta_z^{\tau-1})^\top(\tilde{\boldsymbol{x}}_i^\tau - \boldsymbol{t}^{\tau-1})\| \\ &+\|\Delta\dot{\boldsymbol{x}}_i^\tau - R(\theta_z^{\tau-1})^\top\Delta\tilde{\boldsymbol{x}}_i^\tau\|\,,\end{aligned} \tag{5.7}$$

where the output estimate $\boldsymbol{x}_i^\tau$ is transformed into the world coordinate system by the camera pose estimates consisting of the rotation angle $\theta_z^\tau = \theta_z^{\tau-1} + \Delta\theta_z^\tau$ and 2D translation $\boldsymbol{t}^\tau = R(\Delta\theta_z^\tau)\boldsymbol{t}^{\tau-1} + \Delta\boldsymbol{t}^\tau$.

Observer Reprojection Loss What makes view birdification unique from other on-ground trajectory modeling problems is its ego-centric view input. Although the 2D ego-centric view degenerates depth information of the observed pedestrian movements, it also provides a powerful inductive bias for on-ground trajectory estimates. Using the observer's camera intrinsic matrix A, we compute the reprojection loss in the image plane

$$\mathcal{L}_{\text{proj}} = \sum_{\tau=1}^{T}\sum_{i=1}^{N} \|\overline{\boldsymbol{p}}_i^\tau - sA\overline{\boldsymbol{x}}_i^\tau\|\,, \tag{5.8}$$

where $\overline{\boldsymbol{p}} = \left[p_x, p_y, 1\right]^\top$ is the homogeneous coordinate of the observed 2D bounding box center, and $\overline{\boldsymbol{x}} = [x, y, h/2]^\top$ is the half point of the pedestrian height standing on the position $\boldsymbol{x}_i = R(\Delta\theta_z)\tilde{\boldsymbol{x}}_i + \Delta\boldsymbol{t}$, respectively. The scaling factor s is determined by normalizing the $z-$value of the projected point in the image.

Total Loss The complete multi-task loss becomes

$$\mathcal{L} = \mathcal{L}_{\text{traj}} + \lambda_1\mathcal{L}_{\text{ego}} + \lambda_2\mathcal{L}_{\text{proj}}\,. \tag{5.9}$$

To facilitate stable training, we apply curriculum learning to the reprojection loss weight λ_2. We set $\lambda_2 = 0$ for the first 200 epochs, and switch to $\lambda_2 > 0$ for the rest of the epochs.

Test-time refinement The reprojection loss can be used to refine the ego-motion towards the pedestrian trajectory estimates at inference time. That is, we incor-

porate the reprojection errors into our network as a soft geometric constraint *i.e.,* weighted reprojection loss, in the training phase, and as a hard geometric constraint at inference time.

5.4 Experiments

5.4.1 View Birdification Datasets

We evaluate our method on view birdification dataset consisting of paired real pedestrian trajectories and synthetic ego-views of them. The dataset is generated from public pedestrian trajectory datasets ETH [73] and UCY [50] by following the instructions of the original view birdification paper [66]. To generate a sufficient amount of ego-views including diverse patterns of projected movements, we virtually mount two virtual, perspective cameras in front and rear on each of the pedestrians (*i.e.,* an observer) in turn. Negative heights of bounding boxes indicate the observation from a rear camera. As a result, we obtain paired trajectories and their ego-views for as many as the number of pedestrians in each scene. Following previous work [66], we assume ideal observation, *i.e.,* pedestrians are not occluded by each other and projected heights can be deduced from the observed images. There are three datasets named after the scenes they capture, **Hotel**, **ETH**, and **Students**, which correspond to sparse, moderate, and dense crowds, respectively. We prepare two types of splits of the view birdification dataset. The first one is (i) intra-scene validation split. For each scene, train, val, and test splits are generated. This allows evaluation of how ViewBirdiformer generalizes to unseen trajectories. The second one is (ii) cross-scene validation split. We pick one scene for testing and choose the rest of the remaining scenes for validation and training. These splits allow evaluation of how ViewBirdiformer generalizes to unknown scenes.

Evaluation Metric Our proposed framework first reconstructs the ego-motion of the observer and the trajectories of her surrounding pedestrians in the observer's camera coordinate system. The absolute positions and trajectories of the pedestrians in the world coordinate system are computed by coupling these two outputs, *i.e.,* $\boldsymbol{x}_\pi^\tau = R(\theta_z^{\tau-1} + \Delta\theta_z^\tau)\tilde{\boldsymbol{x}}_i^\tau + R(\Delta\theta_z^\tau)\boldsymbol{t}^{\tau-1} + \Delta\boldsymbol{t}^\tau$. We evaluate the accuracy of our method by measuring the differences of the estimated positions of pedestrians $\boldsymbol{x}$ and the ego-motion of the observer $\Delta\Pi = (\Delta\boldsymbol{t}, \Delta\theta_z)$ from their cor-

responding ground truths $\dot{\boldsymbol{x}}$, $\Delta\dot{\boldsymbol{t}}$, and $\Delta\dot{\theta}_z$. The translation error of the observer is $\Delta\boldsymbol{t} = \frac{1}{T}\sum\|\boldsymbol{x}_\pi^\tau - \dot{\boldsymbol{x}}_\pi^\tau\|$, where T denotes the duration of a sequence. The rotation error of the observer is $\Delta\boldsymbol{r} = \frac{1}{T}\sum_\tau \arccos(\frac{\mathrm{tr}\left(R(\Delta\dot{\theta}_z^\tau)R(\Delta\theta_z^\tau)^\top\right)-1}{2})$, where tr is the matrix trace. We also evaluate the absolute and relative reconstruction errors of pedestrians by $\Delta\boldsymbol{x} = \frac{1}{N}\frac{1}{T}\sum_i\sum_\tau\|\boldsymbol{x}_i^\tau - \dot{\boldsymbol{x}}_i^\tau\|$ and $\Delta\tilde{\boldsymbol{x}} = \frac{1}{N}\frac{1}{T}\sum_i\sum_\tau\|\tilde{\boldsymbol{x}}_i^\tau - R(\theta_z^{\tau-1})\dot{\boldsymbol{x}}_i - \boldsymbol{t}^{\tau-1}\|$.

Baseline Methods We compare our method with a purely geometric view birdification approach [66], the only other view birdification method. We use the parameter values from the original paper, which we refer to as *GeoVB-CV* and *GeoVB-SF* based on the assumed motion model: Constant Velocity (CV) [79] and Social Force (SF) [33], respectively. We also evaluate the effectiveness of the ego-view encoder and the cross-attention by comparing with the direct use of an on-ground motion model which takes $\tau-1$ on-ground trajectories as inputs and simply predicts positions and velocities $\boldsymbol{x}^\tau \oplus \Delta\boldsymbol{x}^\tau$ for τ. For this, we train a simple Transformer-based motion model with one multi-head self-attention layer which we refer to as *TransMotion*. Note that, although *ViewBirdiformer* and *GeoVB* both take as inputs the ego-centric view at time τ and the past on-ground trajectory estimates at time $\tau-1$, *TransMotion* only takes past on-ground trajectory estimates.

We consider two variants of ViewBirdiformer. The first, *ViewBirdiformer-I*, is trained on the intra-scene validation split, and the second, *ViewBirdiformer-C*, on the cross-scene validation split. Similarly, simple motion models composed of single-layer self-attention Transformers each trained with these validation splits are referred to as *TransMotion-I* and *TransMotion-C*, respectively.

Figures 5.4 to 5.6 visualize qualitative results of ViewBirdiformer-I without post-processing from the Hotel, ETH, and Students datasets, respectively. Our ViewBirdiformer outputs sufficient localization accuracy, *i.e.*, <5 [cm] errors in 20×20 [m] fields, for both the ego-motion and surrounding pedestrians for diverse densities of crowds.

Implementation Details All networks were implemented in PyTorch. The camera intrinsic matrix A was set to that of a generic camera with FOV=120° and $f = 2.46$. Both the embedded dimension of the on-ground trajectories and in-image movements, d is set to 32. We use an MLP with 16 hidden units for embedding input features. The number of heads for the multi-head attention layer is all set to 8. Loss coefficient λ_1 is set to 1.0 and λ_2 is set to 0.3 after 200 epochs. We use

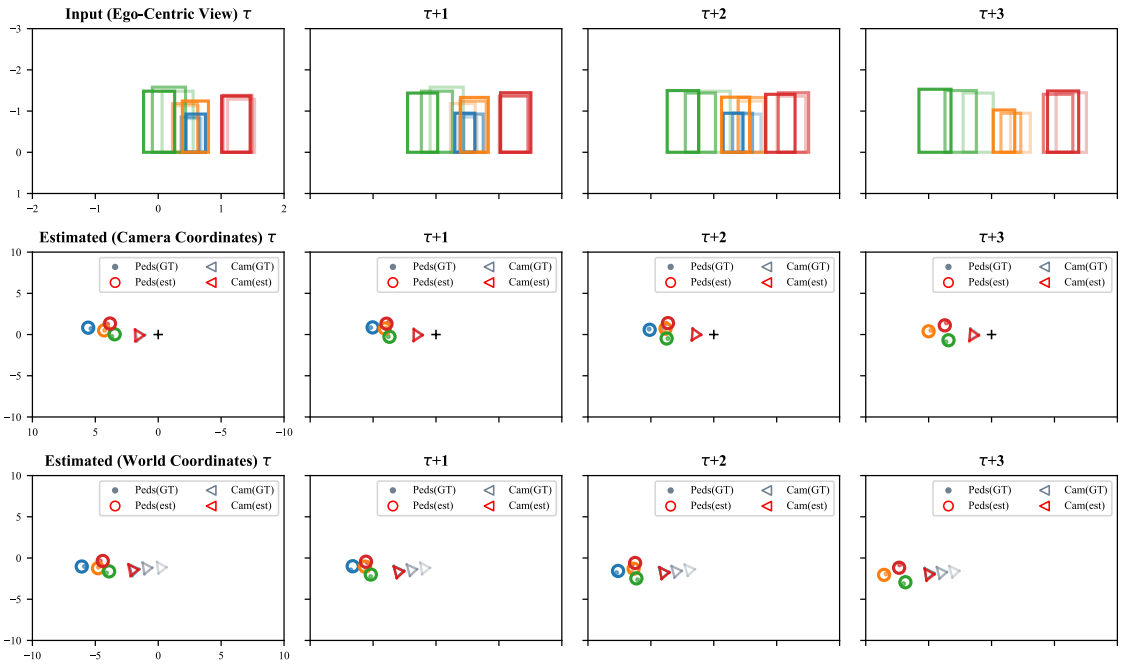


Figure 5.3: **Qualitative Results of ViewBirdiformer-I without post-processing applied to ETH datasets.** The top row shows the input bounding boxes, where the same color box corresponds to the same pedestrian ID and the boxes with low alpha values correspond to the past $\tau - 1$ frame positions. The second row shows the reconstructed camera pose and pedestrian locations at time τ in the $\tau - 1$ camera-centric coordinates. "+" depicts the origin of the camera coordinate system. These relative observations are converted to the world coordinates by the estimated camera pose at every frame (the third row). Grey triangles and circles denote ground-truth camera and pedestrian positions, respectively. 2023 ©IEEE [68]

Adam optimizer and set the constant learning rate to 0.001 for all epochs. All the models are trained with a single NVIDIA Tesla V100 GPU and Intel Xeon Gold 6252 CPU. The training time is approximately 3 hours for the train split excluding Students and 14 hours for that including Students. For all the datasets, we transformed trajectories into scene-centered coordinates so that the origin of the mean position of all the pedestrians is 0. The outputs of our proposed network are post-processed by the test-time refinement described in Sec. 5.3.3.

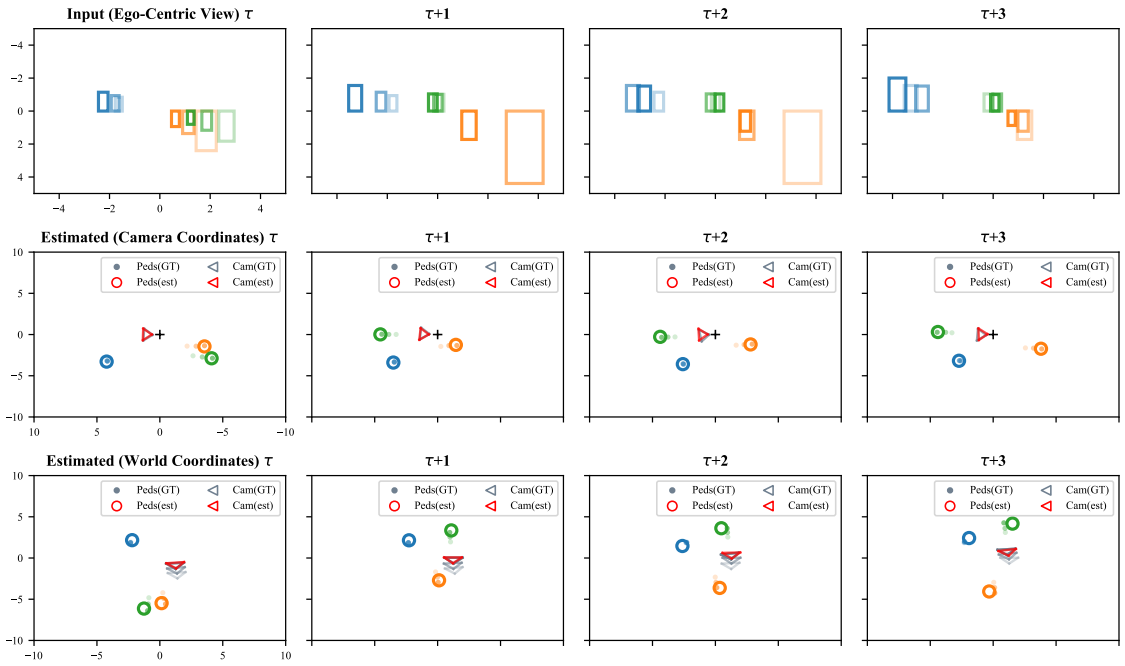


Figure 5.4: **Qualitative Results from the Hotel dataset.** The top row shows detected pedestrian bounding boxes, where the same color corresponds to the same pedestrian ID and boxes with low alpha values correspond to past frame positions. The second row shows the reconstructed camera pose and the pedestrian location in the camera-centric coordinates. These relative observations are converted to world coordinates by the estimated camera pose at every frame (the third row).

5.4.2 Comparison with Geometric Baseline

Localization Accuracy Table 5.2 shows quantitative results. GeoVB [66] achieves high accuracy by iteratively optimizing the camera ego-motion and pedestrian positions by densely sampling possible positions for every frame. Although the accuracy of our ViewBirdiformer is slightly lower, it achieves sufficiently high absolute accuracy (*e.g.*, 5cm errors in 20×20 m field) with a single inference pass. Figure 5.3 visualizes qualitative results of our method on a typical crowd sequence, which clearly shows that our method reconstructs accurate on-ground trajectories. Even with the cross-scene validation split, *ViewBirdiformer-C* achieves comparable results. By incorporating the geometric refinement at inference time, ViewBirdiformer achieves comparable or superior accuracy to the state-of-the-art [66] but still in three orders of magnitude shorter time.

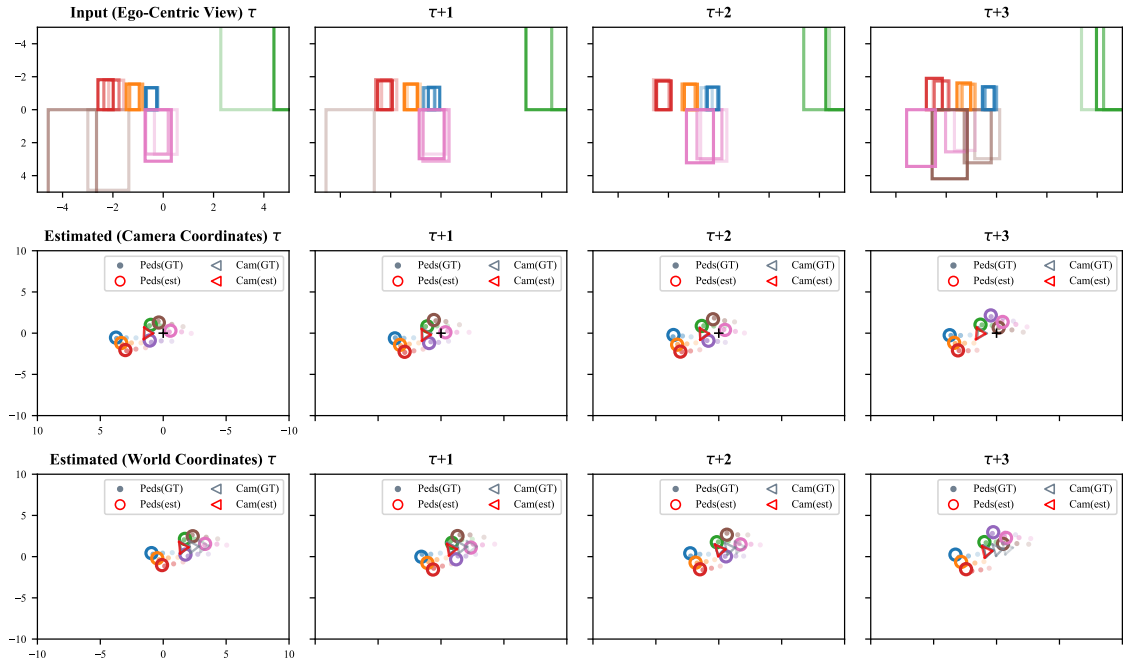


Figure 5.5: **Qualitative Results from the ETH datasets.** Our method can reconstruct global layouts of the camera and its surrounding pedestrians considering interactions between them in moderately crowded scenario. Considering human-to-human and human-to-camera interactions by attention layers, ViewBirdiformer-I recovers on-ground trajectories of surrounding pedestrians while reconstructing the observation camera pose accurately.

Dataset	Hotel / sparse			ETH / mid			Students / dense		
	$\Delta\tilde{x}$ [m]	Δr [rad]	Δt [m]	$\Delta\tilde{x}$ [m]	Δr [rad]	Δt [m]	$\Delta\tilde{x}$ [m]	Δr [rad]	Δt [m]
w/o RelTransform	2.115	0.055	0.277	2.105	0.053	0.279	1.713	0.090	0.269
w/o ReprojectionLoss	0.148	0.148	0.197	0.180	0.038	0.179	0.081	0.065	0.111
Ours	**0.123**	**0.125**	**0.085**	**0.170**	**0.032**	**0.093**	**0.071**	**0.061**	**0.068**

Table 5.1: **Ablation Studies.** w/o denotes our proposed architecture without the specified component. The results demonstrate that relative transformation of the decoder inputs (Section 5.3.2) is essential for accurate localization of surrounding pedestrians, and the additional reprojection loss is key to accurate ego-motion estimation.

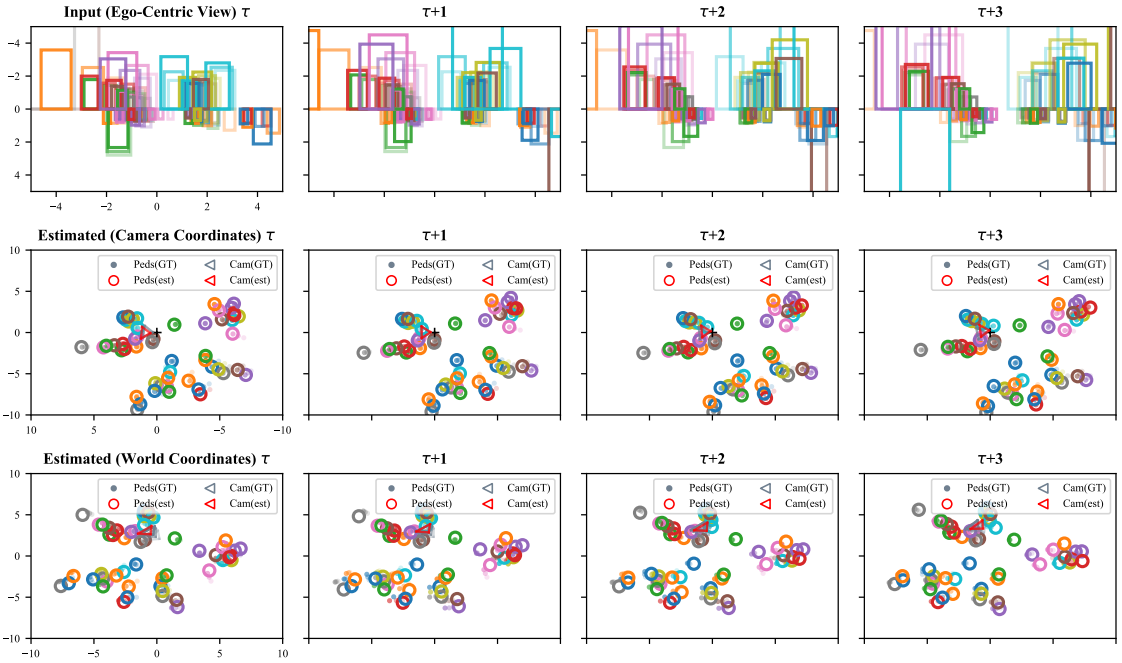


Figure 5.6: **Qualitative Results from the Students datasets.** Our method can accurately reconstruct on-ground trajectories of a large number of pedestrians with a single inference pass, which results in significant inference speedup in densely crowded scenario.

		Hotel / sparse		ETH / mid		Students / dense	
		$\Delta\tilde{x}$ [m]	Δx [m]	$\Delta\tilde{x}$ [m]	Δx [m]	$\Delta\tilde{x}$ [m]	Δx [m]
TransMotion-I		–	0.183	–	0.201	–	0.216
TransMotion-C		–	0.106	–	0.223	–	0.211
GeoVB-CV [66]		**0.051**	0.070	0.089	0.115	0.023	0.024
GeoVB-SF [66]		**0.048***	**0.052***	**0.070***	**0.079***	**0.009***	**0.010***
ViewBirdiformer-I		0.123	0.123	0.170	0.170	0.071	0.071
ViewBirdiformer-C		0.097	0.098	0.216	0.217	0.058	0.059
ViewBirdiformer-I	w/post-processing	0.062	0.081	0.087	0.102	**0.010**	**0.010***
ViewBirdiformer-C	w/post-processing	0.071	0.092	0.099	0.115	**0.010**	**0.011**
		$\Delta\boldsymbol{r}$ [rad]	$\Delta\boldsymbol{t}$ [m]	$\Delta\boldsymbol{r}$ [rad]	$\Delta\boldsymbol{t}$ [m]	$\Delta\boldsymbol{r}$ [rad]	$\Delta\boldsymbol{t}$[m]
GeoVB-CV [66]		**0.015**	0.066	**0.016**	0.095	**0.001***	**0.010**
GeoVB-SF [66]		**0.015**	**0.062**	**0.015***	**0.089**	**0.001***	**0.009***
ViewBirdiformer-I		0.125	0.085	0.032	0.093	0.061	0.068
ViewBirdiformer-C		0.063	0.091	0.101	0.098	0.080	0.069
ViewBirdiformer-I	w/post-processing	**0.014***	**0.059***	**0.015***	**0.091**	**0.002**	**0.011**
ViewBirdiformer-C	w/post-processing	**0.016**	**0.061**	0.021	0.098	**0.002**	**0.011**

Table 5.2: **Quantitative Results.** The top table shows relative and absolute localization errors of pedestrian trajectories, $\Delta\tilde{x}$ and Δx. The motion model baseline only extrapolates the on-ground movement and thus results in missing entries (-) in $\Delta\tilde{x}$. The bottom table shows the camera ego-motion errors Δr and $\Delta\boldsymbol{t}$. We highlight the best (*) and similar to best (accuracy gap $\leq$ 0.005) results of localization accuracy. The results demonstrate the effectiveness of our proposed ViewBirdiformer.

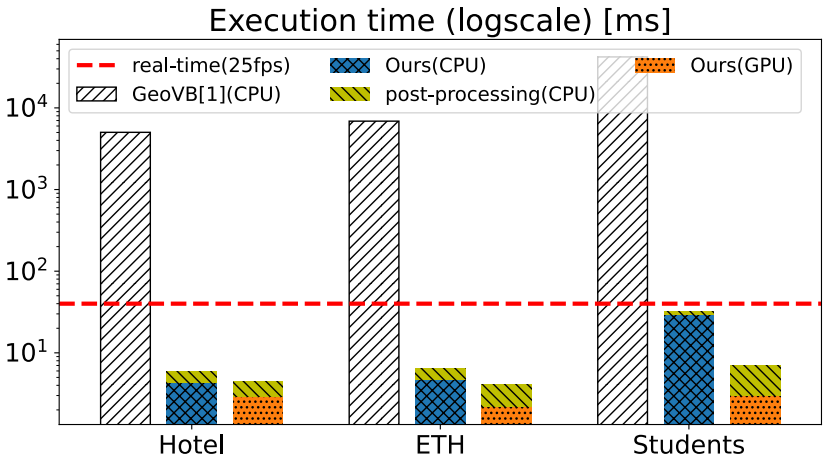


Figure 5.7: **Execution time.** We measure the execution times of our method on a CPU and a GPU. The post-processing is executed on the CPU. These results are averaged over the samples.

Execution Time Figure 5.7 shows the execution time of our method and GeoVB [66] on a single Intel Core i5-7500 CPU and a NVIDIA GeForce 1080Ti GPU. These results clearly demonstrate the efficiency of our method compared to GeoVB. The unified transformer architecture of our method enables estimation of both ego-motion and pedestrian trajectories with a single inference pass without the costly iteration process in GeoVB [66], which results in three orders of magnitude improvement in execution time. For N pedestrians, S samples, and T iterations, the computational complexity of GeoVB is $\mathcal{O}(NS^2T)$ and it is hardly parallelizable as it requires sequential update over all possible samples S ($S \gg N$). In contrast, the computational complexity of ViewBirdiformer is $\mathcal{O}(N^2d)$ [85] and its implementation can naturally be parallelized within GPU, which collectively realize this significant reduction in execution time. Most important, even with the geometric refinement at inference time, ViewBirdiformer achieves accuracy on par with the state-of-the-art [66] while maintaining this orders of magnitude faster execution.

5.4.3 Ablation Studies

Cross-Attention Between Views Table 5.2 compares the accuracy of ViewBirdiformer and simple extrapolation of on-ground movements using dedicated simple transformers. While ViewBirdiformer takes the current ego-centric view and the past on-ground trajectory estimates as inputs, TransMotion only takes the

past trajectory estimates as inputs. ViewBirdiformer shows superior performance over TransMotion in pedestrian localization. These results clearly show that the cross-attention mechanism between on-ground motions and movements in the ego-centric views is essential for accurate trajectory estimation of the surrounding pedestrians.

Relative Position Transformation Table 5.1 shows the results of ablating the relative position transforms (Section 5.3.2). All models are trained with the intra-scene split of the birdification dataset to avoid generalization errors of the learnt motion model. *w/o RelTransform* takes on-ground trajectories in world coordinates as decoder inputs. Without the relative position transformations described in Sec. 5.3.1, the proposed framework shows significant accuracy drops, especially in pedestrian localization. This is likely caused by the inconsistency of the coordinate system between on-ground past trajectory inputs and egocentric view inputs and demonstrates the importance of the relative transformation for generalization of the model.

Reprojection Loss *w/o ReprojectionLoss* in Tab. 5.1 considers only the ego-motion loss and the pedestrian trajectory loss, *i.e.,* $\lambda_2 = 0$ in eq. (5.9). The results show that the reprojection loss slightly improves the accuracy of ego-motion estimates. This is because the reprojection loss works similarly to geometric constraints as in *GeoVB*.

5.4.4 Limitations and Degenerate Scenario

If the observed relative movements are static (*i.e.,* an observer is following the pedestrian at the same speed), our model cannot break the fundamental ambiguity. Such degenerate scenarios, however, rarely happen in crowds as there will be other pedestrians. Our method also assumes that the heights of pedestrians are more or less the same and that the detected bounding boxes are correct. We plan to relax these requirements by developing an end-to-end framework that handles both tracking and birdification while estimating uncertainty in depth arising from variances of pedestrians.

5.5 Implementation Details

Weights of the Multi-Task Loss One critical choice of our multi-task formulation in Sec.4.3 is the weight λ_1 for the ego-motion loss relative to the trajectory loss, which balances the accuracy trade-offs in localization for the observer and surrounding pedestrians. For this, we investigate the localization accuracy for different values of $\lambda_1 \in \{0.2, 0.5, 1.0, 2.0, 3.0\}$ using the validation data split. Figure 5.8 shows the results of ViewBirdiformer-I trained on the ETH dataset with different ego-motion loss weights λ_1. These results demonstrate that as the weight of the ego-motion loss λ_1 increases, ego-motion accuracies (Δr, $\Delta \boldsymbol{t}$) improve, while the accuracy in the relative pedestrian localization $\Delta\tilde{\boldsymbol{x}}$ degrades. In ViewBirdiformer, these two are affected by each other in test-time refinement (Sec.4.3), which requires accurate estimation of both the ego-motion and pedestrian trajectories. We thus selected the value of lambda $\lambda_1 = 1.0$ that optimally trades-off between the ego-motion estimation (Δr, $\Delta \boldsymbol{t}$) and pedestrian localization ($\Delta\tilde{\boldsymbol{x}}$).

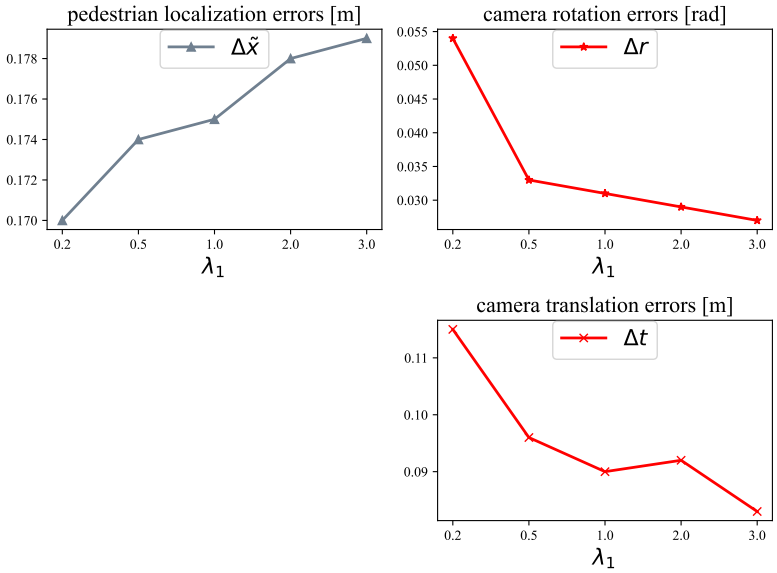


Figure 5.8: **Localization accuracy for several different choices of λ_1.** The left column shows the errors in pedestrian localization $\Delta\tilde{\boldsymbol{x}}$ [m]. The right two columns show the errors in camera localization (rotation Δr [rad] and translation $\Delta \boldsymbol{t}$ [m]). Typically, as the weights of ego-motion λ_1 increases, the errors of the pedestrian localization increase, while the errors of rotation and translation decrease. 2023 ©IEEE [68]

Table 5.3: Localization errors of **ViewBirdiformer-I** tested with view birdification dataset w/o occlusion and w/ occlusion (**visible pedestrian only**). ViewBirdiformer-I w/ occlusion shows comparable accuracy even considering occlusions.

Dataset		Hotel (sparse)			ETH (mid)			Students (dense)		
		$\Delta\tilde{x}$ [m]	Δr [rad]	Δt [m]	$\Delta\tilde{x}$ [m]	Δr [rad]	Δt [m]	$\Delta\tilde{x}$ [m]	Δr [rad]	Δt [m]
Ours w/o	occlusion	0.123	0.125	0.085	0.170	0.032	0.093	0.071	0.061	0.068
Ours w/	occlusion	0.129	0.143	0.087	0.174	0.035	0.101	0.071	0.064	0.066

5.6 Handling Occluded Pedestrians

Our transformer model handles varying numbers of pedestrians coming in and out of frames, some of whom may be occluded in practice. In order to learn the crowd motion model from all the pedestrians on the ground, we train our model using the view birdification dataset without occlusions [66]. To validate the applicability of our model in realistic scenarios, we extend the view birdification dataset [66] by simulating occlusions between pedestrians. As in Sec.4.1 "*Attention Mask*" of the main text, our model can handle occluded pedestrians by simply masking the attention matrix for those pedestrians without any changes to the algorithm. Table 5.3 compares the localization accuracy of ViewBirdiformer-I tested with the dataset with occlusions (**w/ occlusion**) against that on the non-occluded dataset (**w/o occlusion**). For **w/ occlusion** dataset, we birdify only non-occluded pedestrians and evaluate localization errors of them, by masking attention to the occluded pedestrian ids. These results clearly show that our model pretrained on the synthetic, non-occluded dataset can handle realistic occlusions with only a mild performance drop.

5.7 Comparison of Crowd Motion Models

We compare the localization accuracy of the learned crowd motion model in the self-attention (*i.e.,* TransMotion) with prescribed motion models (*i.e.,* ConstVel [79] and SocialForce [33] used in GeoVB [66]). As shown in Tab. 5.4, the results clearly show that our TransMotion can estimate future location of pedestrians more accurately than the predetermined models. While ConstVel and SocialForce perform poorly with Hotel and ETH since the models are not capable of approximating these sparse interactions, our ViewBirdiformer can explain diverse interactions in the scene regardless of the densities of crowds. Our model

	Hotel (sparse) Δx [m]	**ETH** (mid) Δx [m]	**Students** (dense) Δx [m]
ConstVel [79]	0.294	0.275	0.223
SocialForce [33]	0.289	0.261	0.222
TransMotion-I	**0.183**	**0.201**	**0.216**
TransMotion-C	**0.106**	**0.223**	**0.211**

Table 5.4: The localization errors of pedestrian trajectories Δx for each motion model. Compared to the ConstVel and SocialForce models which perform poorly in sparsely crowded environments, our learned motion model shows superior performance over diverse crowd densities.

trained with leave-one-out validation (*i.e.*, TransMotion-C) also demonstrates that the learned motion model can be generalized to diverse densities of crowds without training data for the target scene. This implies our method can be applied to real-world crowds with complex pedestrian interactions.

5.8 Attention Visualization

Figure 5.9 visualizes the learned attention in the multi-head attention layers of ViewBirdiformer for typical example sequences. The left column visualizes it for camera ego-motion estimation. The right column visualizes it for the trajectory estimation of one of the pedestrians. In cross-attention between the ego-centric view and the on-ground trajectories, we can see that our model adaptively attends both to the ego-centric view and the on-ground past trajectories to better predict the next-step trajectories conditioned on the ego-centric movements. ViewBirdiformer attends to all pedestrians when estimating the camera ego-motion, while it attends to pedestrians nearby the target, likely to avoid collisions, when estimating the position of the pedestrian.

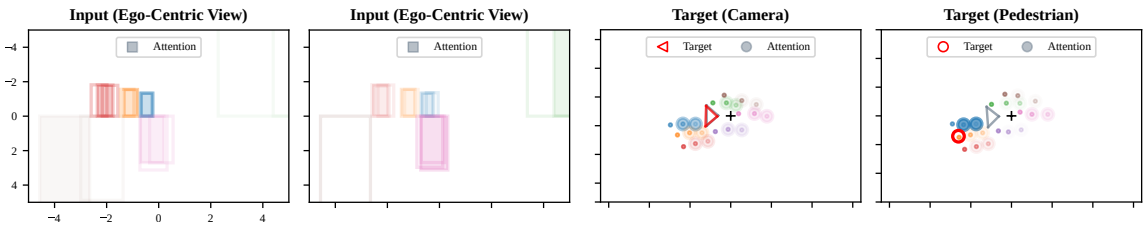


Figure 5.9: **Attention Visualization.** The left column visualizes attention weights on the ego-centric view inputs (top) and those on the on-ground past trajectories (bottom) when estimating the ego-motion. The right column visualizes the attention weights when estimating the position of the target pedestrian. The more opaque, the stronger the attention.

Chapter 6

Pedestrian World Model

6.1 Background

Everyday, we successfully maneuver in a highly dynamic world with just our egocentric, limited view of the surroundings. When we walk down the street, we constantly map the positions of surrounding pedestrians on the ground plane. We not only keep track of their current positions, but also predict their next positions. When we play football, for instance, we are able to tell where other players will end up, which is exactly why we can make that killer pass.

This predictive mental model of our dynamic surroundings is an illustrative example of "World Models" [27], transition models of the environment. In this chapter, we are particularly interested in deriving a world model of pedestrians that can continuously localize all visible surrounding people and predict their movements in the next few time steps. An accurate world model of people from our egocentric views will enable efficient and accurate modeling of our people-filled dynamic world and benefit various tasks including navigation, tracking, and synthesis of crowds.

A world model of pedestrians that can be useful for such downstream tasks, however, requires significant departures from past models [27, 44, 14]. First, it must model the pedestrians on the ground plane but from inputs in egocentric perspectives. This requires a nontrivial view transformation, often referred to as Bird's-Eye-View (BEV) transform. The world model, however, needs to predict the next on-ground movements from the observed current movements in the 2D egocentric view, unlike BEV transform that focuses on an image to image transform (*i.e.*, appearance mapping) of only the current frame. We refer to this inherently predictive purely geometric transform as ego2top transform.

Second, the model needs to be fundamentally object-centric. It must model each pedestrian as an independent object that interacts with other objects including the ego-viewer, too. The interaction between pedestrians lies at the heart of the coordination of the pedestrians as a whole, and the transitions of individual pedestrians are largely governed by these object interactions especially in denser crowds. Third, the model needs to model and predict the pedestrian movements and their interactions conditioned on the observer. The observer itself is part of the crowd, and its/his/her actions influence and are affected by the surrounding people. Finally, the viewer would be moving and looking around while walking in the very crowd that needs to be modeled. As a result, surrounding people will come in and out of sight. The crowd itself will also consist of different numbers of people from time to time. The model thus needs to naturally handle a varying number of people.

We derive a world model of pedestrians that satisfy these requirements, namely ego2top transform, object-centric interaction encoding, observer action conditioned modeling, and variable number of constituents. We refer to our model as *InCrowdFormer*. The key idea is to model pedestrians as individual tokens and fully leverage attention for modeling their interactions and ego2top transform in a Transformer architecture which also naturally models varying numbers of pedestrians. Unlike previous approaches [20, 14] which model spatial encoding and temporal prediction separately, our unified Transformer architecture simultaneously encodes the social, temporal, and geometric relationships of the viewer and surrounding pedestrians.

Two key challenges underlie learning an accurate Pedestrian World Model. The model needs to decouple the ego-motion and pedestrian trajectories from degenerated 2D movements in the image. In addition, the unknown absolute scale of each object, *i.e.*, pedestrian heights, introduces uncertainty in the ego2top transform. For this, we introduce a latent code for each pedestrian and construct a generative model that outputs the on-ground future location distribution conditioned on the observed movements in an ego-centric view and the observer's action. We efficiently model this conditional probability with a Transformer consisting of self-attention that encodes the social and temporal relationship between the observer's action and each pedestrian, and cross-attention that models the geometric relationship between the ego-centric and on-ground views. We generate plausible observer trajectories [84] for training, which naturally lets the model learn transitions of the world when navigating in a crowd while avoiding in-

tractable numbers of combinations of actions and crowd motions.

InCrowdFormer only requires pedestrian bounding boxes in the egocentric view, which allows us to train it without access to actual images and overcome the domain gap between synthetic and real views and also across scenes. We validate the effectiveness of our *InCrowdFormer* with prediction and navigation data of real-world pedestrian movements by synthetically generating ego-centric views of crowd pedestrians but of ground-truth trajectories extracted from publicly available crowd datasets [50, 73]. Extensive experimental evaluation show that our unified Transformer World Model can accurately predict the future coordination of pedestrians given the observer's action while taking into account uncertainty arising from imperfect cues of depths. We also demonstrate the application of our pretrained model to real video sequences.

In summary, our contributions are threefold. (1) We introduce, to our knowledge, the first egocentric Pedestrian World Model that models pedestrian and in-crowd observer transitions on the ground plane from egocentric observations, (2) derive it as a novel Transformer that leverages attention for pedestrian interaction modeling and view transform for a variable number of people, and (3) demonstrate its accuracy on real-world crowd motions. We believe our *InCrowdFormer* will serve as a sound foundation of pedestrian movement modeling for a wide range of applications. We will release our code and data upon acceptance.

6.2 Pedestrian World Model

Our goal is to derive an on-ground *Pedestrian World Model* (PWM), an object-oriented abstraction of a crowd on the ground, from egocentric views captured in the crowd. PWM consists of one observer as an actor and pedestrians visible to that observer as objects. We derive an action-conditioned transition model of the environment consisting of these objects that can directly learn from in-environment 2D egocentric perception.

Consider a mobile robot equipped with a vision sensor immersed in a crowd consisting of people walking towards their own destinations while interacting with each other. A key characteristic of the PWM we aim to derive is that, unlike past object-oriented world models [54], the observer is the actor and is also part of the environment who interacts with other objects (pedestrians). In our running example, the mobile robot with a vision sensor is the observer and its ego-motion can be obtained by an IMU or other sensors including vision-based

methods (*e.g.,* SLAM [9] and View Birdification [66]). The observer location at τ is given by the relative rotation $R(\theta_\tau) \in SO(2)$ and translation $\boldsymbol{t}_\tau \in \mathbb{R}^2$ on the ground plane from the previous timestep $\tau - 1$. The rotation $R(\theta_\tau)$ and $\boldsymbol{t}_\tau$ directly constitute the observer action $\boldsymbol{a}_\tau = [\theta_\tau | \boldsymbol{t}_\tau]^\top \in \mathbb{R}^3$.

At every timestep $\tau \in \{1, \dots, T\}$, the robot captures an image I_τ through which it observes the in-image states $\boldsymbol{X}_\tau = \{\boldsymbol{x}^1, \dots, \boldsymbol{x}^{N_\tau}\}$ of N_τ people visible from the robot. Each state $\boldsymbol{x}^n = [u^n, v^n, \delta u^n, \delta v^n]^\top$ $(n = 1, \dots, N_\tau)$ consists of the 2D center position $[u^n, v^n]^\top$ and its velocity $[\delta u^n, \delta v^n]^\top$ in I_τ calculated from the previous and subsequent frames. These states can be computed from the images with an off-the-shelf multi-object tracker [90], and the robot can keep track of $\{1, \dots, N_\tau\}$ pedestrians appearing across frames within a time window centered at τ. Our objective is to predict future on-ground states of pedestrians $\boldsymbol{Y}_{\tau+1} = \{\boldsymbol{y}^1, \dots, \boldsymbol{y}^{N_\tau}\}$, where $\boldsymbol{y}^n = [x^n, y^n, \delta x^n, \delta y^n]^\top$. The on-ground pedestrian states $\boldsymbol{y}$ are described in the observer's camera coordinates, *i.e.,* we predict 2D on-ground locations and their velocities relative to the observer's view which is then converted to absolute coordinates with the known observer state.

Given a set of in-image pedestrian states $\boldsymbol{X}_\tau$ at the current time step τ and the action $\boldsymbol{a}_\tau$ of the observer, our aim is to construct a transition model $\mathcal{T}(\boldsymbol{X}_\tau | \boldsymbol{a}_\tau) \mapsto \boldsymbol{Y}_{\tau+1}$ that predicts the on-ground pedestrian states $\boldsymbol{Y}_{\tau+1} \in \mathbb{R}^{N_\tau \times 4}$ in the future conditioned on the observer's action $\boldsymbol{a}_\tau$. We can formulate this as learning a transition model

$$\boldsymbol{H}_\tau = \mathcal{V}(\boldsymbol{X}_\tau), \quad \boldsymbol{Y}_{\tau+1} = \mathcal{G}(\boldsymbol{H}_\tau, \boldsymbol{Y}_\tau, \boldsymbol{a}_\tau), \tag{6.1}$$

where $\mathcal{V}(\cdot)$ is the Vision Module that encodes the ego-centric view observation $\boldsymbol{X}_\tau \in \mathbb{R}^{N_\tau \times 4}$ into a d-dimensional embedding $\boldsymbol{H}_\tau \in \mathbb{R}^{N_\tau \times d}$ which represents a set of pedestrian state embeddings in an ego-centric view, and $\mathcal{G}(\cdot)$ is the Geometric Memory Module which makes predictions of the future on-ground pedestrian states $\boldsymbol{Y}_{\tau+1}$ based on the past on-ground state estimates $\boldsymbol{Y}_\tau$ and the current observation $\boldsymbol{X}_\tau$ with an implicit transform between the ego-centric view and the top-down ground view.

The mapping $\mathcal{V}$ from the in-image pedestrian states $\boldsymbol{X}_\tau$ to the on-ground pedestrian states $\boldsymbol{H}_\tau$ should take into account the relationship between the pedestrians. Similarly, the mapping $\mathcal{G}$ should model the interaction between the pedestrians and the observer. We leverage the learnable set-to-set mapping of the attention mechanism [85] which is also inherently agnostic to varying numbers of inputs, *i.e.,* the number of pedestrians N_τ observed by the robot.

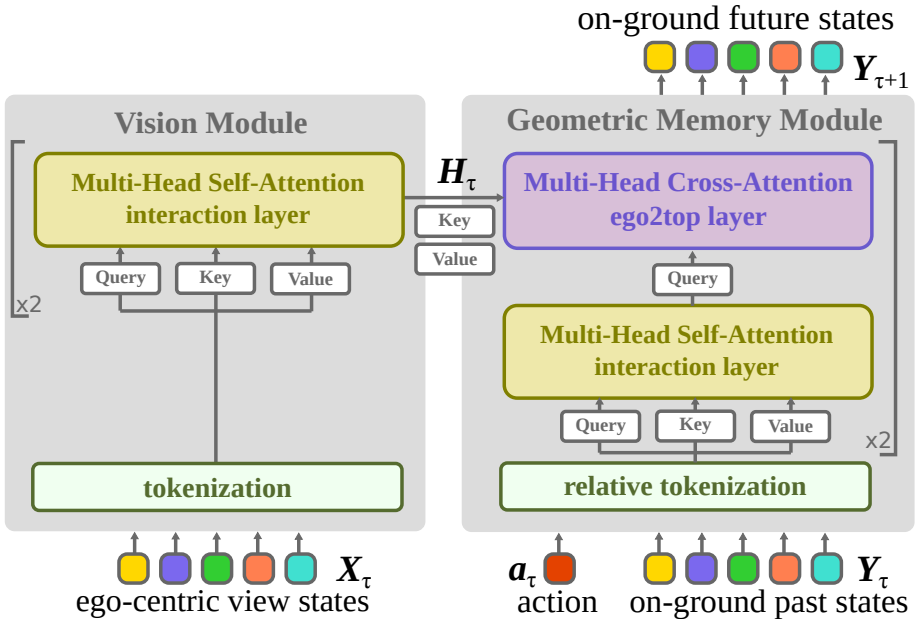


Figure 6.1: InCrowdFormer consists of two modules referred to as the Vision Module and the Geometric Memory Module. The Vision Module encodes in-image interactions with self-attention to produce the current state embeddings $\boldsymbol{H}_\tau$ in an ego-centric view. The Geometric Memory Module predicts on-ground future states of pedestrians with cross-attention between the ego-centric view and on-ground past trajectories.

6.3 InCrowdFormer

We introduce InCrowdFormer, a Transformer-based World Model, that realizes a PWM with object-centric interaction, ego2top transform, action conditioning, for a variable number of pedestrians. As fig. 6.1 depicts, InCrowdFormer has two modules, the Vision Module $\mathcal{V}$ and the Geometric Memory Module $\mathcal{G}$.

6.3.1 Vision Module

The attention mechanism naturally provides a means to learn a mapping between two sets with variable numbers of constituents. We fully leverage this to learn the key ingredients, namely pedestrian interaction and ego2top transform both naturally conditioned on the observer actions, of an on-ground Pedestrian World Model from egocentric views.

The standard attention mechanism consists of a set of query vectors $\boldsymbol{Q}$, key vectors $\boldsymbol{K}$, and value vectors $\boldsymbol{V}$. These vectors are generated from the input tokens

$\boldsymbol{f}_s$ and $\boldsymbol{f}_t$ as

$$\boldsymbol{Q} = W_q \boldsymbol{f}_t\,, \quad \boldsymbol{K} = W_k \boldsymbol{f}_s\,, \quad \boldsymbol{V} = W_v \boldsymbol{f}_s\,, \tag{6.2}$$

where W_Q, W_K, and W_V are linear embedding matrices. The attention mechanism is

$$\text{Attn}_{(\boldsymbol{Q}_i, \boldsymbol{K}_j, \boldsymbol{V}_j)} = \sum_j A_{ij} \boldsymbol{V}_j\,, A_{ij} = \frac{\exp(\boldsymbol{Q}_i^\top \boldsymbol{K}_j)}{\sum_{j'=1}^{N} \exp(\boldsymbol{Q}_i^\top \boldsymbol{K}_{j'})}\,, \tag{6.3}$$

where we apply a softmax as a nonlinearity layer. In *self-attention* $\boldsymbol{f}_s = \boldsymbol{f}_t$ and for *cross-attention* we have $\boldsymbol{f}_s \neq \boldsymbol{f}_t$. Unless otherwise noted, we use multi-head attention [85] for both self-attention and cross-attention layers.

To model the interactions between pedestrians in the current frame, we use self-attention to learn the object-wise interactions observed in the ego-centric view during the state encoding process. As fig. 6.1 left depicts, the Vision Module $\mathcal{V}$ first tokenizes input first-person-view (FPV) state vectors into d_s-dimensional tokens with multi-layer perceptron (MLP) layers, $\boldsymbol{X}_\tau \mapsto \boldsymbol{f}_s \in \mathbb{R}^{N_\tau \times d_s}$. It then computes self-attention over queries $\boldsymbol{Q}$ of tokens $\boldsymbol{f}_s$, where we encode their interactions in the image space into intermediate embeddings $\boldsymbol{H}_\tau$.

6.3.2 Geometric Memory Module

Learning a Pedestrian World Model is fundamentally different from standard trajectory forecasting problems [98] in that the on-ground movements of a crowd that needs to be predicted is deeply intertwined with the observer's ego-motion (*i.e.,* action). That is, how the pedestrians move relative to the observer is largely affected by the observer's action. For this, we model a mapping $\mathcal{G}(\boldsymbol{a}, \boldsymbol{H}_\tau, \boldsymbol{Y}_\tau) \mapsto \boldsymbol{Y}_{\tau+1}$ by $\mathcal{G} = \mathcal{F}_c(\mathcal{F}_s(\boldsymbol{a}_\tau, \boldsymbol{Y}_\tau), \boldsymbol{H}_\tau)$, where we use self-attention $\mathcal{F}_s$ to capture social interactions and cross-attention $\mathcal{F}_c$ to model the ego2top transform. This Geometric Memory Module first tokenizes the on-ground action and past pedestrian states as $\{\boldsymbol{a}_\tau, \boldsymbol{Y}_\tau\} \mapsto \boldsymbol{f}_t \in \mathbb{R}^{(N_\tau+1)\times d_s}$ with MLPs and computes self-attention over queries $\boldsymbol{Q}$ from tokens $\boldsymbol{f}_t$ to encode the interactions between the action and on-ground pedestrian trajectories. The cross-attention block takes queries $\boldsymbol{Q} \in \mathbb{R}^{(N_\tau+1)\times d_s}$ from the output of the self-attention layer and key, values $\boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{N_\tau \times d_s}$ from the output of the vision module (Figure 6.1 left).

The geometric memory module is autoregressive, which means the module predicts future on-ground states one step at a time and uses the current prediction as input to make future predictions on a subsequent timestep. As the FPV

view state embeddings $\boldsymbol{H}_\tau$ are updated by the vision module $\mathcal{V}$ at every timestep, their on-ground future predictions $\boldsymbol{Y}_{\tau+1}$ are generated by the geometric memory module $\mathcal{G}$ from the past predictions $\boldsymbol{Y}_\tau$, the current observer's action $\boldsymbol{a}_\tau$, and the FPV state embeddings $\boldsymbol{H}_\tau$.

We use two linear projections; $\varphi_p : \mathbb{R}^4 \mapsto \mathbb{R}^{d_s}$ for encoding in-image and on-ground pedestrian states, $\boldsymbol{X}$ and $\boldsymbol{Y}$, and $\varphi_a : \mathbb{R}^3 \mapsto \mathbb{R}^{d_s}$ for the observer's action $\boldsymbol{a}$. We hard-concatenate positional information into states as $\boldsymbol{x} = [u, v] \oplus [\delta u, \delta v]$. To encourage our model to generalize over diverse combinations of pedestrian tokens and observer actions, we transform on-ground pedestrian tokens relative to the observer's action token at every timestep.

InCrowdFormer can handle a varying number of pedestrians, *i.e.,* tokens, in two ways. First, the attention matrix can be constructed with arbitrary sizes of input keys $\boldsymbol{K}$ and queries $\boldsymbol{Q}$. Second, an attention mask $M \in \mathbb{R}^{(N_\tau+1)\times N_\tau}$, where $M_{ij} = 0$ can be applied, if the pedestrian id i is missing (*e.g.,* occluded) at frame τ, otherwise 1. The masked attention $\hat{A}$ becomes

$$\hat{A}_\tau = M_\tau \odot A_\tau \,, \tag{6.4}$$

where $\odot$ denotes the Hadamard product.

6.4 Probabilistic InCrowdFormer

We encode uncertainties arising from unknown pedestrian heights by making InCrowdFormer reason probabilistically on the pedestrian positions and their transitions. Figure 6.2 depicts an overview of the training and testing process of our Probabilistic InCrowdFormer.

To model the distribution of the future on-ground states of pedestrians $p(\boldsymbol{Y}_{\tau+1}|\boldsymbol{X}_\tau, \boldsymbol{a}_\tau)$ conditioned on the ego-centric view observation $\boldsymbol{X}_\tau$ and the observer's action $\boldsymbol{a}_\tau$, we formulate it as a conditional variational autoencoder (CVAE). We introduce d_z-dimensional latent codes for each pedestrian $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{N_\tau}\} \in \mathbb{R}^{N_\tau \times d_z}$ and re-formulate the future distribution of the pedestrian states

$$p(\boldsymbol{Y}_{\tau+1}|\boldsymbol{X}_\tau, \boldsymbol{a}_\tau) = \int \underbrace{p(\boldsymbol{Y}_{\tau+1}|\boldsymbol{Z}, \boldsymbol{X}_\tau, \boldsymbol{a}_\tau)}_{\text{likelihood}} \underbrace{p(\boldsymbol{Z}|\boldsymbol{X}_\tau)}_{\text{prior}} d\boldsymbol{Z} \,, \tag{6.5}$$

where $p(\boldsymbol{Y}_{\tau+1}|\boldsymbol{Z}, \boldsymbol{X}_\tau, \boldsymbol{a}_\tau)$ is the conditional likelihood and $p(\boldsymbol{Z}|\boldsymbol{X}_\tau) = \prod_n p(\boldsymbol{z}^n|\boldsymbol{X}_\tau)$

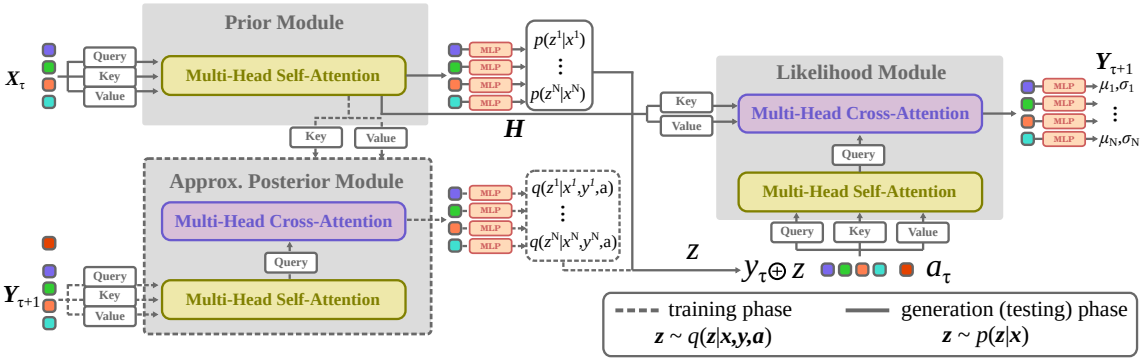


Figure 6.2: Overview of Probabilistic InCrowdFormer. The output MLP layer of the Vision Module models the prior distribution. Each of the two geometric memory modules models the approximated posterior distribution and likelihood distribution, respectively. The latent codes are sampled from the approximated posterior module during training, and from the prior module at inference time.

is the conditional Gaussian prior factorized over pedestrians. The observer's action is deterministic in our Pedestrian World Model, so that latent codes are only necessary for pedestrians and not the observer.

Due to the intractable integral computation in eq. (6.5), we minimize the negative evidence lower bound (ELBO) in our loss function $\mathcal{L}$

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\boldsymbol{X}_\tau, \boldsymbol{Y}_{\tau+1}, \boldsymbol{a}_\tau; \eta, \phi, \psi) = \\ \mathbb{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{Y},\boldsymbol{X})}\left[\log \underbrace{p_\eta(\boldsymbol{Y}_{\tau+1}|\boldsymbol{Z}, \boldsymbol{X}_\tau, \boldsymbol{a}_\tau)}_{\text{likelihood}}\right] \\ - \text{KL}\left[\underbrace{q_\phi(\boldsymbol{Z}|\boldsymbol{Y}_{\tau+1}, \boldsymbol{X}_\tau, \boldsymbol{a}_\tau)}_{\text{approximated posterior}} \| \underbrace{p_\psi(\boldsymbol{Z}|\boldsymbol{X}_\tau)}_{\text{prior}}\right],\end{aligned} \tag{6.6}$$

where the first term maximizes the expectation of the log-likelihood of the target future state in the predicted distribution, and the second term minimizes the Kullback-Leibler (KL) divergence between the approximated posterior distribution and the prior distribution.

Given the current observed pedestrian states in the ego-centric view $\boldsymbol{X}_\tau$, the observer's action $\boldsymbol{a}_\tau$, and future pedestrian state relative to the observer $\boldsymbol{Y}_{\tau+1}$, our goal is to learn a generative model $p(\boldsymbol{Y}_{\tau+1}|\boldsymbol{X}_\tau, \boldsymbol{a}_\tau)$ by learning network parameters η. In what follows, we describe how we implement each distribution in our

Probabilistic InCrowdFormer.

Prior Module The *Vision Module* can be seen as as a prior network that produces latent codes $\boldsymbol{Z} = \{z^1, \ldots, z^{N_\tau}\}$ each of which follows a Gaussian distribution $p(\boldsymbol{z}^n|\boldsymbol{X}_\tau) \sim \mathcal{N}(\mu^n, (\sigma^n)^2)$. We process the output of the Vision Module $\boldsymbol{H}_\tau = \{\boldsymbol{h}^1, \ldots, \boldsymbol{h}^{N_\tau}\}$ with a single MLP layer to map them into parameters of each Gaussian distribution (μ_p^n, σ_p^n).

Approximated Posterior Module To approximate the posterior distribution $q_\phi(\boldsymbol{z}^n|\boldsymbol{Y}_{\tau+1}, \boldsymbol{X}_\tau, \boldsymbol{a}_\tau)$, we use a Transformer decoder with the same self-attention and cross-attention layers as the *Geometric Memory Module*. The cross-attention layers in the Transformer decoder can efficiently model conditional relationships by taking the embeddings of FPV states $\boldsymbol{H}_\tau$ as keys and values, and the action and the ground-truth future state on the ground $\{\boldsymbol{a}_\tau, \boldsymbol{Y}_{\tau+1}\}$ as queries. We also assume the approximated posterior distribution follows a Gaussian distribution $q_\phi(\boldsymbol{z}^n|\boldsymbol{Y}_{\tau+1}, \boldsymbol{X}_\tau, \boldsymbol{a}_\tau) \sim \mathcal{N}(\mu^n, (\sigma^n)^2)$. MLP layers process the output of the Transformer Decoder to obtain the Gaussian parameters (μ_q^n, σ_q^n) as in the Prior Module. Note that this approximated posterior module is used only during training.

Likelihood Module We can view the *Geometric Memory Module* as a likelihood module to model $p_\eta(\boldsymbol{Y}_{\tau+1}|\boldsymbol{Z}, \boldsymbol{X}_\tau, \boldsymbol{a}_\tau)$. The latent codes $\boldsymbol{Z}$ for pedestrians are computed from the approximated posterior module during training, and from the vision module for inference. As we obtain predictions of future pedestrian states autoregressively from the geometric memory module, we feed $\boldsymbol{Y}$ concatenated with corresponding latent codes $\boldsymbol{Z}$ alongside the observer's action $\boldsymbol{a}_\tau$. We concatenate $\boldsymbol{y}^n \oplus \boldsymbol{z}^n$ for each pedestrian and tokenize these as input queries. The cross-attention layers in the geometric memory module effectively models the conditional relationship between $\boldsymbol{Z}$ and $\boldsymbol{a}_\tau$ and outputs a likelihood distribution. We consider two variants of the proposed method based on the definition of the outputs, **InCrowdFormer-D** and **InCrowdFormer-G**.

InCrowdFormer-D directly estimates the future pedestrian states and, for the first term of eq. (6.6), uses the mean squared error (MSE), $\mathcal{L}_\mathrm{D} = \|\hat{\boldsymbol{Y}} - \boldsymbol{Y}\|^2$, where $\hat{\boldsymbol{Y}}$ is the ground-truth future state of pedestrians on the ground.

InCrowdFormer-G takes into account uncertainties arising from ego2top transformation of unknown object scales as aleatoric uncertainty [42] computed as a probability distribution over the InCrowdFormer outputs. We model the uncertainty with a 2D Gaussian distribution on the ground $p(\boldsymbol{Y}_{\tau+1}|\boldsymbol{Z}, \boldsymbol{X}_\tau, \boldsymbol{a}_\tau) \sim \mathcal{N}(\boldsymbol{\mu}^n, \Sigma^n)$, and make InCrowdFormer output these Gaussian parameters for each

pedestrian by defining the first term of eq. (6.6) as a negative log-likelihood loss of a 2D Gaussian function $\mathcal{L}_{\mathrm{G}} = \frac{\|\hat{\boldsymbol{Y}}_x - \boldsymbol{Y}_x\|^2}{(2\sigma_x)^2} + \frac{\|\hat{\boldsymbol{Y}}_y - \boldsymbol{Y}_y\|^2}{(2\sigma_y)^2}$. Such a parametric representation of uncertainty of the future pedestrian state would be useful for downstream applications.

6.5 Experiments

We evaluate the effectiveness of Probabilistic InCrowdFormer (hereafter simply InCrowdFormer) trained on real-world crowd trajectories with augmented observer's actions in terms of its T-step prediction accuracy and also demonstrate its application to real video sequences.

Dataset (IV)	Hotel (sparse)				ETH (mid)				Students (dense)			
Metrics	ADE_5	FDE_5	ADE_{10}	FDE_{10}	ADE_5	FDE_5	ADE_{10}	FDE_{10}	ADE_5	FDE_5	ADE_{10}	FDE_{10}
TSSM [20, 14]	4.009	4.763	6.014	6.432	4.102	4.835	5.986	6.287	5.939	6.126	6.723	6.921
RSSM [27]	4.251	4.861	6.027	6.853	4.562	4.925	6.102	6.489	6.102	6.615	6.532	6.911
InCrowdFormer-D	0.358	0.468	0.626	0.829	0.326	0.385	0.537	0.683	0.251	0.273	0.432	0.483
InCrowdFormer-G	0.327	0.472	0.648	0.818	0.372	0.393	0.526	0.652	0.312	0.372	0.461	0.492
Dataset (CV)	Hotel (sparse)				ETH (mid)				Students (dense)			
TSSM [20, 14]	4.228	4.934	6.100	6.353	4.147	4.941	6.175	6.318	6.064	6.208	6.737	7.007
RSSM [27]	4.315	4.856	6.223	6.843	4.549	5.020	6.105	6.615	5.849	6.649	6.544	7.002
InCrowdFormer-D	0.687	0.638	0.717	1.037	0.469	0.421	0.580	0.601	0.323	0.383	0.512	0.568
InCrowdFormer-G	0.536	0.272	0.714	0.881	0.372	0.428	0.646	0.722	0.437	0.523	0.578	0.720

Table 6.1: Quantitative Results of InCrowdFormer applied to the ego-centric view crowd dataset. (IV) and (CV) indicate the intra-scene, and the cross-scene validation split, respectively. ADE_T and FDE_T represent average and final displacement errors [m] in T-step prediction. InCrowdFormer variants clearly outperform baseline methods in terms of 5-step and 10-step prediction accuracy. Prediction results applied to the cross-scene validation split dataset demonstrate that our method can be generalized to unknown crowd scenarios.

Ego-centric View Crowd Dataset We construct a crowd dataset consisting of on-ground real pedestrian trajectories paired with their state features in an ego-centric view. We first extract trajectories referred to as **Hotel**, **ETH**, and **Students** from the ETH [73] and the UCY [50] datasets. These three sets of trajectories correspond to sparse, moderate, and dense crowds. To compute the egocentric views of those pedestrians, we first sample pedestrian heights from a Gaussian distribution $\mathcal{N}(\mu, \sigma)$ with $\mu = 1.70$ m and $\sigma = 0.07$ m according to the statistics of European adults [87]. The virtual head points with the height $h \sim \mathcal{N}(\mu, \sigma)$ are then projected to the 2D positions $[u, v]$ by perspective projection with a known intrinsic matrix A. For each pedestrian, we compute the state feature $[u, v, \delta u, \delta v]$ in the ego-centric view and pair it with its corresponding on-ground position. Since our proposed Pedestrian World Model is an abstraction of the on-ground crowd movements, we do not need photo-realistic renderings in the datasets. This allows us to augment datasets with diverse combinations of ego-motion and crowd pedestrian trajectories easily and efficiently, which is otherwise challenging, if not impossible, to collect in the real world.

Observer Action Generation To model the transition of the world in the observer's view when navigating in a crowd while avoiding potential collisions, we generate plausible observer's trajectories with an Optimal Reciprocal Collision Avoidance (ORCA) planner [84] in the ego-centric view crowd dataset. We randomly sample starting positions on a circle with a fixed radius $r = 8.0$ m and the observer walks to its destination set at the opposite side of the circle by the planner. We mount two virtual, perspective cameras in front and rear on the observer, each of which captures states of the pedestrians in the crowd.

Baselines We compare two variants of world models consisting of a standard MLP object encoder and state-of-the-art memory modules referred to as Recurrent State Space Model (RSSM) [27] and Transformer State Space Model (TSSM) [14]. We refer to these baselines as **MLP-RSSM** and **MLP-TSSM**. In their original implementations, these world models encode in-image transitions, unlike our model which encodes on-ground transitions conditioned on the ego-views. To apply these models to our crowd dataset, we change the object decoder output from the in-image state features to the on-ground state features. We consider two variants of InCrowdFormer, **InCrowdFormer-D** (deterministic) and **InCrowdFormer-G** (Gaussian) as introduced in Sec. 6.4.

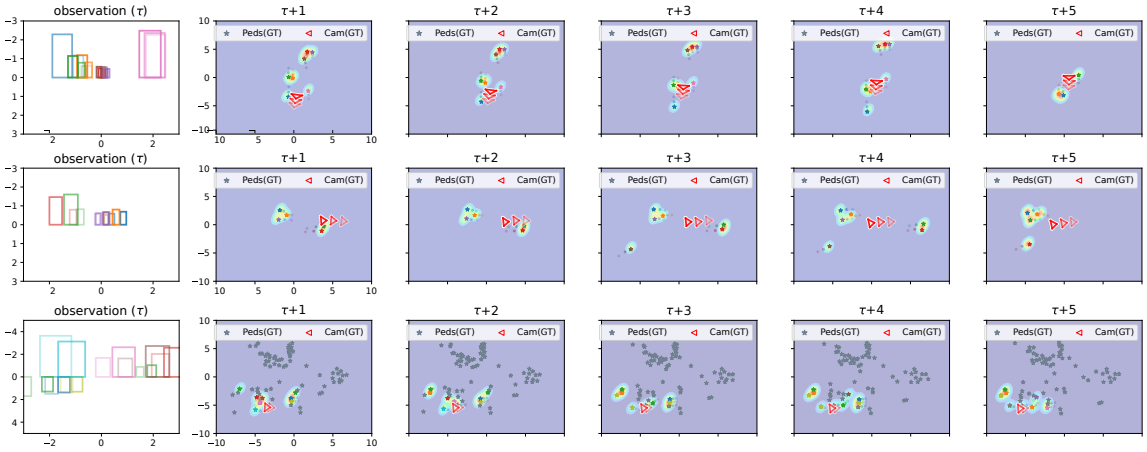


Figure 6.3: Qualitative results of InCrowdFormer-G applied to the Hotel (top row), ETH (middle), and Students (bottom) datasets. Left most: First frames in the ego-centric views. Rest: On-ground future prediction results in subsequent frames. Negative heights of bounding boxes depict observations from the rear camera. The probabilities are rendered with red (high) to blue (low) heatmaps. Stars depict the ground-truth positions. Our model successfully predicts on-ground pedestrian states for crowds of diverse densities. Even in a dense crowd, our method can successfully handle the varying number of pedestrians and predict accurate future locations of nearby pedestrians.

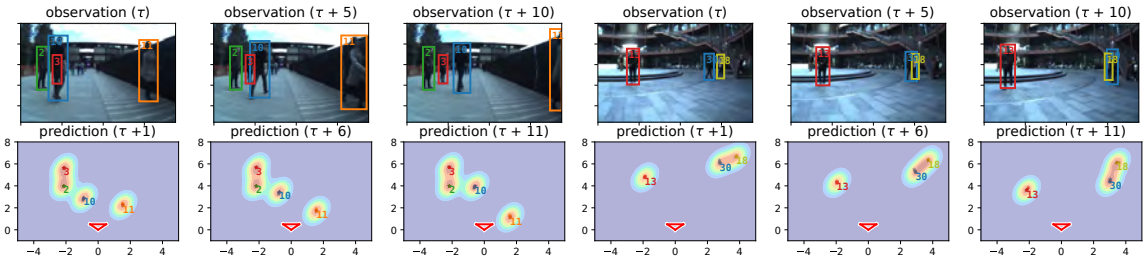


Figure 6.4: Inference results of InCrowdFormer-G applied to real video sequences from the JRDB dataset. Top row shows input ego-centric view video and pedestrian bounding boxes. Bottom row shows on-ground $\tau + 1$ future pedestrian trajectories predicted by our model. Our method can easily be adapted to real video sequences as it only requires the positions and scales of the bounding boxes of pedestrians.

Metric Given a sequence of observer's actions and corresponding ego-centric view observations, we predict the on-ground state of pedestrians in the future by autoregressively feeding an action $\boldsymbol{a}_\tau$ and ego-centric view states $\boldsymbol{X}_\tau$ into our InCrowdFormer model. For a set of action sequences $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_T\}$ and corresponding ego-centric view states $\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_T\}$ during T-steps in the testing set, we evaluate the prediction accuracy with Average Displacement Error (ADE) and Final Displacement Error (FDE). We define $\mathrm{ADE}_T = \frac{1}{TN^T} \sum_{n=1}^{N^T} \sum_{\tau=1}^{T} \|\hat{\boldsymbol{y}}_\tau^n - \boldsymbol{y}_\tau^n\|$ and $\mathrm{FDE}_T = \frac{1}{N^T} \sum_{n=1}^{N^T} \|\hat{\boldsymbol{y}}_T^n - \boldsymbol{y}_T^n\|$, respectively. For InCrowdFormer-D and InCrowdFormer-G, we compute the best-of-10 results, *i.e.,* the minimum ADE and FDE from 10 randomly-sampled latent codes as in [26].

6.5.1 Quantitative evaluation of prediction accuracy

We prepare two validation splits for the ego-centric view crowd dataset. One is the intra-scene validation split, and the other is the cross-scene validation split, *i.e.,* leave-one-out cross-validation. Table 6.1 shows the prediction accuracy evaluated in ADE_T and FDE_T with $T \in \{5, 10\}$. The RSSM baseline makes a prediction without taking object interactions into account, which results in the lowest accuracy in all the metrics. The TSSM can be considered as a subset of our model without a cross-attention mechanism between ego-centric and top-down views. Both the InCrowdFormer-D and InCrowdFormer-C outperform TSSM in terms of short-term ($T = 5$) and long-term ($T = 10$) prediction accuracy. These results clearly demonstrate the effectiveness of our cross-attention mechanism to reconstruct and predict on-ground future pedestrian states. Finally, our method achieves high accuracy on the dataset with the cross-scene validation split. This demonstrates that our method can be generalized to unknown scenes by training the model with a diverse density of crowds. Overall, InCrowdFormer-D achieves slightly higher accuracy than InCrowdFormer-G, which can be attributed to the fact that InCrowdFormer-D directly regresses future positions while InCrowdFormer-G predicts uncertainty distributions. Most important, our method still achieves high (< 1.0 [m]) accuracy for long-term ($T = 10$) prediction sequences as well as short-term ($T = 5$) sequences by attending to observed ego-centric view features at every timestep.

Qualitative Results Figure 6.3 visualizes the prediction results of InCrowdFormer-G from time τ to $\tau + 5$ frames on the Hotel, ETH, and Students datasets. InCrowdFormer-G outputs the future location of a pedestrian with a

2D Gaussian distribution which captures uncertainty arising from imperfect cues of depths and pedestrian interactions. Our method also predicts future locations of nearby pedestrians in a dense crowd by masking the attention matrix for occluded pedestrians. These outputs are beneficial to downstream applications such as robot crowd navigation, where we should path-plan to avoid potential collisions with nearby pedestrians from an ego-centric view and limited depth information.

6.5.2 Inference on Real Data

Figure 6.4 shows the inference results of InCrowdFormer-G applied to real video sequences from the JRDB dataset [58]. We first train our model with the same camera parameter of JRDB. We train on the ETH dataset, which is similar in crowd density to the JRDB sequences. We then apply our pre-trained model to the target sequences. Our method can easily be adapted to real video sequences as it requires only the position and scale of pedestrian bounding boxes and abstracts away their appearance.

Chapter 7

Conclusion

In this dissertation, we introduced *view birdification*, a task of simultaneously recovering the location of a camera in the crowd and its surrounding pedestrians only from perceived movements in an ego-centric video. The key research question lies in the task is *how we can achieve localization and prediction just from dynamic objects*. To answer this question, we first formulated view birdification as a geometric reconstruction problem and extended it as an object-oriented world model. Technically, we derived a foundation based on the Transformer architecture simultaneously to learn geometric transformation and the motion model of dynamic objects, which are the key common challenges in both localization and prediction from in-crowd views. We believe view birdification will become essential for mobile robot navigation and localization in real-world crowds.

7.1 Summary of Contributions

Chapter 3: View Birdification We first formulated view birdification as a geometric reconstruction problem, where we reconstruct 2D on-ground displacements of the observation camera and its surrounding pedestrians from the perceived 2D movements in an ego-centric view. Assuming that trajectories are all on the ground plane, we simplify the complicated localization in dynamic environments as a 2D-to-2D transformation problem. The definition is not restricted to the specific camera model and can be applied to any type of camera models whose projection model is known a priori.

Chapter 4: View Birdification from a Bayesian Perspective In Chapter3, we formulated view birdification as a geometric trajectory reconstruction problem.

The key difficulty underlying this task is that the two kinds of trajectories, the camera ego-motion and pedestrian trajectories on the ground plane, are deeply intertwined in the observed movements in an ego-centric view. To address this, we derived a cascaded optimization from a Bayesian perspective that alternately updates the estimated camera ego-motion and pedestrian locations relative to it. We empirically analyzed the properties of the solution with respect to the number of pedestrians in the crowd, which brings us key insights to further extend this research topic. Our extensive evaluation demonstrates the effectiveness of our proposed view birdification method for crowds of varying densities. We believe view birdification has implications for both computer vision and robotics, including crowd behavior analysis, self-localization, and situational awareness, and opens new avenues of application including dynamic surveillance.

Chapter 5: Learning to recover ground-plane crowd trajectories and ego-motion In Chapter 4, we derived a cascaded optimization approach for view birdification from a Bayesian perspective and empirically analyzed the property of its solution. However, we also found two critical problems with this approach. First, the formulation assumes a known motion model, which heavily restricts applications to the real-world crowd. Second, the computational cost of this iterative approach is too large to achieve localization in real-time. To address these problems, we proposed a data-driven framework to efficiently obtain the solution by learning from a pair of trajectories of pedestrians and an observation camera. We refer to this as a ViewBirdiformer. The proposed architecture enables efficient and accurate view birdification by adaptively attending to movement features of the observer and pedestrians in the image plane and on the ground. Extensive evaluations demonstrate the effectiveness of ViewBirdiformer for crowds with diverse pedestrian interactions. We believe ViewBirdiformer finds use in various applications of crowd modeling and synthesis across a wide range of disciplines.

Chapter 6: Pedestrian World Model Lastly, we extended view birdification as an object-oriented world model. Unlike conventional world models that predict the future state of the whole image from an egocentric viewpoint, our aim is to construct an object-oriented world model that predicts the future state of each pedestrian while learning the interaction between them. We refer to this object-oriented world model as the Pedestrian World Model, a computational transition model of pedestrians that can continuously localize and predict the movements

of all people visible to the observer. To represent the Pedestrian World Model, we extend view birdification as a transition model that predicts the future state of the crowd from in-crowd views. We derived InCrowdFormer, a Transformer-based Pedestrian World Model that predicts on-ground pedestrian trajectories from an ego-centric view observation. Our extensive evaluation demonstrates the effectiveness of our approach in diverse density of crowds, and also shows promising results in zero-shot adaptation to real video sequences. We believe our InCrowdFormer will serve as a sound foundation for crowd and pedestrian movement modeling and enable a wide range of downstream applications including but not limited to navigation.

7.2 Towards Autonomous Navigation in the Real World

In this dissertation, we focused on establishing a foundation of view birdification under the assumptions that people in a crowd follow a common motion model. We also assume that the observer also moves smoothly in the crowd and the mounted camera does not change its orientation against the observer's moving direction. These assumptions are not realistic and we still have several missing pieces to apply our work to the real-word. Ideally, we would train on real videos of pedestrians. This, however, proves difficult, especially in the current COVID 19 pandemic, as we are not allowed to gather many people in one place. Preservation of privacy also makes filming people on the street difficult. We plan to overcome this dilemma by developing a framework that can synthesize walking motions for real trajectories in a photorealistic simulator, which we believe will benefit the community at large.

Realistic Observer's Movements We assume that the observer also follows the crowd flow in the scene and never changes its orientation of the mounted camera drastically across the frames. Especialyl in data-driven approaches (Sections 5–6), we cannot augment infinite number of combinations of the observer's ego-motion and on-ground trajectories of surrounding pedestrians. Collecting real-world crowd dataset from an ego-centric view (*i.e.*, mobile robot or pedestrians) at scale will be beneficial to analyze the effect of observation noises arising from such realistic movements.

Applications to Diverse Scenarios Our overall evaluations highly depend on publicly available crowd datasets [73, 50]. While these datasets include diverse densities of crowds, the crowd behavior is limited to the simple scenario where people in the crowd just walk towards a fixed destination. In our real world, pedestrians in a crowd have multiple destinations such as shops and buildings and their walking paths are affected not only by interactions between nearby pedestrians but also by semantic scene contexts. We plan to extend our work to introduce the semantic context by encoding the semantic map before calcurating attention between pedestrian tokens.

Integration to Mobile Robot Navigation Platform We do not intend to fully replace the existing static keypoints-based approaches by view birdification. These two approaches are complementary to each other. As discussed in Section 4.3, if the static keypoints are available near the observation camera, we can use static keypoints-based localization rather than view birdification. We suppose that the navigation system seamlessly switch static keypoints-based approaches and dynamic keypoints-based view birdification according to the observed situation. Another promising approach is to fuse multiple localization resources including view birdification according to its reliability in the scene. We hope our dynamic keypoints-only approach expands traversable areas of robots in the dynamic world, where prior approaches easily get lost.

7.3 Future Directions

Fig. 7.1 depicts an overview of our contribution and its relevant applications. View Birdification is a newly defined task and thus there is a lot of room for further research in terms of real-world applications. In what follows, we will discuss possible future research directions.

End-to-end Learnable Framework for Raw Image Inputs In this dissertation, we assumed that pedestrian bounding boxes are detected by external trackers and they are perfectly identified across frames. Also, occlusion handling is carried out by an external multi-object tracker. We envision a feedback loop from our birdificaiton framework that can inform the multi-object tracker to reason better about the occluded targets, which will likely enhance the accuracy as a whole even in heavily occluded scenes. View Birdification can work complementary to

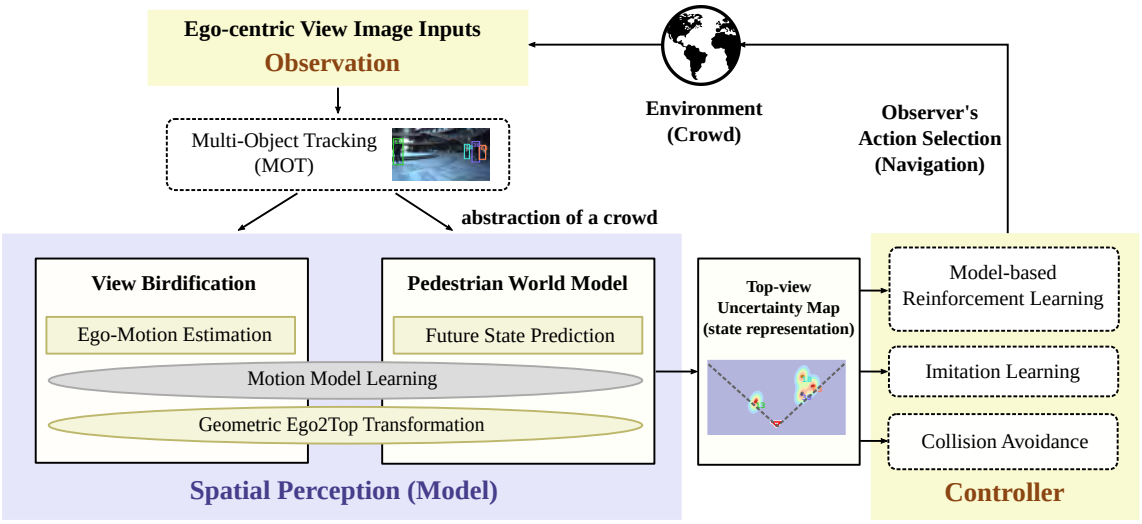


Figure 7.1: Future Directions. We build a foundation of on-ground pedestrian movement estimation and prediction from ego-centric in-crowd views. Our work can be used for various downstream applications such as navigating mobile robots in a crowd. The learned ego-motion estimator and world models will be used as a versatile state representation for in-crowd observations.

3D Multi-Object Tracking [90, 36]. The on-ground ego-motion and the pedestrian motion model which are both estimated by view birdification are indeed helpful to detect and track pedestrians across frames efficiently in the 3D space.

Pedestrian World Model for Uncertainty-aware Navigation One critical limitation of monocular visual perception is uncertainty in depth, which cannot be resolved in principle. Pedestrian World Model and its uncertainty-aware outputs are designed for downstream applications such as crowd-aware mobile robot navigation [69, 13], *e.g.*, the controller plan the observer's next action to avoid potential collision, assuming the tail of the future state distribution as an inflated radius of pedestrians. As illustrated in Fig. 7.1 Right, predicted on-ground future states of pedestrians can be used as inputs for arbitrary controllers to determine the next action of the observer. Specifically, the learnt pedestrian world model can be used as a simulator of the world in Model-Based Reinforcement Learning (MBRL) [89, 63, 4]. Once the model of a crowd conditioned on the observer's movement is learned, we can train the observer's controller only inside of the learnt Pedestrian World Model.

Another interesting direction is simultaneously to estimate localization uncertainty while also estimating the relative location of the pedestrians. In Section 6,

we assumed perfect odometry observation for the observer's action, *i.e.,* known observer ego-motion, and did not consider observation noise in the ego-centric views. We believe that this can be modeled similarly as uncertainty in depth arising from pedestrian's height variances, which we plan to explore in future work.

Transformer as a Policy Network Conventional reinforcement learning-based navigation approaches design the transition model and the policy network with a modular, separated network [27, 20]. On the contrary, Transformers recently achieved remarkable success in reinforcement learning by formulating Markov Decision Process (MDP) as a sequence modeling problem [15, 61]. Specifically, Transformers learn to predict the next best action by taking attention over the previous states, actions, and rewards. Given a sequence of demonstrations, *e.g.,* a pair of states and corresponding actions, Transformer-based architecture can take long-term dependencies into account effectively and efficiently. Inspired by this, one possible research direction is to extend our Transformer-based Pedestrian World Model as a policy as it can naturally predict the ego-motion conditioned on the surrounding pdestrian's movements. We believe Transformers will become a powerful foundation to model and translate an object-oriented world while learning the interactions between them.

Bibliography

[1] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Nonrigid structure from motion in trajectory space. In *Proc. NeurIPS*, 2008.

[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proc. CVPR*, pages 961–971, 2016.

[3] Bani Anvari and Helge A. Wurdemann. Modelling social interaction between humans and service robots in large public spaces. In *Proc. IROS*, pages 11189–11196, 2020.

[4] Mohammad Babaeizadeh, Mohammad Taghi Saffar, Danijar Hafner, Harini Kannan, Chelsea Finn, Sergey Levine, and Dumitru Erhan. Models, pixels, and rewards: Evaluating design trade-offs in visual model-based reinforcement learning. *arXiv preprint arXiv:2012.04603*, 2020.

[5] Vijay Badrinarayanan, Ignas Budvytis, and Roberto Cipolla. Mixture of trees probabilistic graphical model for video segmentation. *IJCV*, 110(1):14–29, 2014.

[6] Irene Ballester, Alejandro Fontán, Javier Civera, Klaus H. Strobl, and Rudolph Triebel. Dot: Dynamic object tracking for visual slam. In *Proc. ICRA*, pages 11705–11711, 2021.

[7] Steven Bell, Alejandro Troccoli, and Kari Pulli. A non-linear filter for gyroscope-based video stabilization. In *Proc. ECCV*, pages 294–308. Springer, 2014.

[8] Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi. Monoloco: Monocular 3d pedestrian localization and uncertainty estimation. In *Proc. ICCV*, pages 6861–6871, 2019.

[9] Berta Bescos, José M Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018.

[10] Pierre-Andre Brousseau and Sebastien Roy. Calibration of axial fisheye cameras through generic virtual central models. In *Proc. ICCV*, 2019.

[11] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qizhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *Proc. ECCV*. 2020.

[12] Changan Chen, Sha Hu, Payam Nikdel, Greg Mori, and Manolis Savva. Relational graph learning for crowd navigation. In *Proc. IROS*, 2020.

[13] Changan Chen, Yuejiang Liu, Sven Kreiss, and Alexandre Alahi. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *Proc. ICRA*, pages 6015–6022. IEEE, 2019.

[14] Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

[15] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Proc. NeurIPS*, 34:15084–15097, 2021.

[16] Yuying Chen, Congcong Liu, Bertram E Shi, and Ming Liu. Robot navigation in crowds by graph convolutional networks with attention learned from human gaze. *IEEE Robotics and Automation Letters*, 5(2):2754–2761, 2020.

[17] Anthony Cioppa, Adrien Deliege, Floriane Magera, Silvio Giancola, Olivier Barnich, Bernard Ghanem, and Marc Van Droogenbroeck. Camera calibration and player localization in soccernet-v2 and investigation of their representations for action spotting. In *Proc. CVPR*, pages 4537–4546, 2021.

[18] Kenneth James Williams Craik. *The nature of explanation*, volume 445. CUP Archive, 1967.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. ICLR*, 2020.

[20] Daniel Dugas, Olov Andersson, Roland Siegwart, and Jen Jen Chung. NavDreams: Towards camera-only RL navigation among humans. In *Proc. IROS*, 2022.

[21] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Proc. ECCV*, pages 834–849. Springer, 2014.

[22] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006.

[23] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[24] Jay W Forrester. Counterintuitive behavior of social systems. *Theory and decision*, 2(2):109–140, 1971.

[25] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Proc. NeurIPS*, 33:1970–1981, 2020.

[26] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proc. CVPR*, pages 2255–2264, 2018.

[27] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Proc. NeurIPS*, pages 2451–2463. Curran Associates, Inc., 2018.

[28] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *Proc. ICLR*, 2019.

[29] Danijar Hafner, Timothy P Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proc. ICML*, 2019.

[30] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *Proc. ICLR*, 2021.

[31] Dirk Hahnel, Rudolph Triebel, Wolfram Burgard, and Sebastian Thrun. Map building with mobile robots in dynamic environments. In *Proc. ICRA*, volume 2, pages 1557–1563. IEEE, 2003.

[32] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *TPAMI*, 2022.

[33] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[34] Mina Henein, Jun Zhang, Robert Mahony, and Viorela Ila. Dynamic slam: The need for speed. In *Proc. ICRA*, pages 2123–2129. IEEE, 2020.

[35] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. FIERY: Future instance segmentation in bird's-eye view from surround monocular cameras. In *Proc. ICCV*, 2021.

[36] Hou-Ning Hu, Yung-Hsu Yang, Tobias Fischer, Trevor Darrell, Fisher Yu, and Min Sun. Monocular quasi-dense 3d object tracking. *TPAMI*, 2022.

[37] Jiahui Huang, Sheng Yang, Tai-Jiang Mu, and Shi-Min Hu. Clustervo: Clustering moving instances and estimating visual odometry for self and surroundings. In *Proc. CVPR*, pages 2168–2177, 2020.

[38] Elwan Héry, Philippe Xu, and Philippe Bonnifait. Distributed asynchronous cooperative localization with inaccurate gnss positions. In *Proc. ITSC*, pages 1857–1863, 2019.

[39] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proc. ICCV*, pages 2375–2384, 2019.

[40] Sebastian Hoppe Nesgaard Jensen, Mads Emil Brix Doest, Henrik Aanaes, and Alessio Del Bue. A benchmark and evaluation of non-rigid structure from motion. *IJCV*, 2020.

[41] Philip Nicholas Johnson-Laird. *Mental models: Towards a cognitive science of language, inference, and consciousness*. Number 6. Harvard University Press, 1983.

[42] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Proc. NeurIPS*, 30, 2017.

[43] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[44] Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *Proc. ICLR*, 2019.

[45] Louis Kratz and Ko Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *Proc. CVPR*, pages 1446–1453. IEEE, 2009.

[46] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Openpifpaf: Composite fields for semantic keypoint detection and spatio-temporal association. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):13498–13511, 2022.

[47] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Multi-body non-rigid structure-from-motion. In *Proc. 3DV*, pages 148–156, 2016.

[48] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proc. ICML*, pages 3744–3753. PMLR, 2019.

[49] Kuan-Hui Lee, Kliemann Matthew, Gaidon Adrien, Li Jie, Fang Chao, Pillai Sudeep, and Burgard Wolfram. Pillarflow: End-to-end birds-eye-view flow estimation for autonomous driving. In *Proc. IROS*, 2020.

[50] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Computer graphics forum*, 26(3):655–664, 2007.

[51] Jose Lezama, Karteek Alahari, Josef Sivic, and Ivan Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *Proc. CVPR*, pages 3369–3376, 2011.

[52] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. *Proc. AAAI*, 2023.

[53] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *Proc. ECCV*, 2022.

[54] Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn. Improving generative imagination in object-centric world models. In *Proc. ICML*, 2020.

[55] Jonathon Luiten, Tobias Fischer, and Bastian Leibe. Track to reconstruct and reconstruct to track. *IEEE Robotics and Automation Letters*, 5(2):1803–1810, 2020.

[56] Yan Luo, Chongyang Zhang, Muming Zhao, Hao Zhou, and Jun Sun. Where, what, whether: Multi-modal learning meets pedestrian detection. In *Proc. CVPR*, pages 14065–14073, 2020.

[57] Osama Makansi, Özgün Çiçek, Kevin Buchicchio, and Thomas Brox. Multimodal future localization and emergence prediction for objects in egocentric view with a reachability prior. In *Proc. CVPR*, pages 4354–4363, 2020.

[58] Roberto Martin-Martin, Mihir Patel, Hamid Rezatofighi, Abhijeet Shenoi, JunYoung Gwak, Eric Frankel, Amir Sadeghian, and Silvio Savarese. Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[59] Gerrit W Maus, Jason Fischer, and David Whitney. Motion-dependent representation of space in area mt+. *Neuron*, 78(3):554–562, 2013.

[60] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *Proc. CVPR*, pages 935–942. IEEE, 2009.

[61] Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *Deep Reinforcement Learning Workshop NeurIPS 2022*.

[62] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[63] Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

[64] T. Moore and D. Stouch. A generalized extended kalman filter implementation for the robot operating system. In *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, July 2014.

[65] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[66] Mai Nishimura, Shohei Nobuhara, and Ko Nishino. View birdification in the crowd: Ground-plane localization from perceived movements. In *Proc. BMVC*, 2021.

[67] Mai Nishimura, Shohei Nobuhara, and Ko Nishino. View birdification in the crowd: Ground-plane localization from perceived movements. *CoRR*, abs/2111.05060, 2021.

[68] Mai Nishimura, Shohei Nobuhara, and Ko Nishino. Viewbirdiformer: Learning to recover ground-plane crowd trajectories and ego-motion from a single ego-centric view. *IEEE Robotics Autom. Lett.*, 8(1):368–375, 2023.

[69] Mai Nishimura and Ryo Yonetani. L2b: Learning to balance the safety-efficiency trade-off in interactive crowd-aware robot navigation. In *Proc. IROS*, pages 11004–11010, 2020.

[70] Mai Nishimura12, Shohei Nobuhara, and Ko Nishino. View birdification in the crowd: Ground-plane localization from perceived movements. 2021.

[71] David Nistér. An efficient solution to the five-point relative pose problem. *TPAMI*, 26(6):756–770, 2004.

[72] Hyun Soo Park, Eakta Jain, and Yaser Sheikh. 3d social saliency from head-mounted cameras. *Proc. NeurIPS*, 25:422–430, 2012.

[73] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proc. ICCV*, pages 261–268, 2009.

[74] Yuheng Qiu, Chen Wang, Wenshan Wang, Mina Henein, and Sebastian Scherer. Airdos: Dynamic slam benefits from articulated objects. In *Proc. ICRA*, pages 8047–8053, 2022.

[75] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[76] Rockstar Games. `https://www.rockstargames.com`.

[77] Avishkar Saha, Oscar Mendez, Chris Russell, and Richard Bowden. Translating images into maps. In *Proc. ICRA*, pages 9200–9206. IEEE, 2022.

[78] Muhamad Risqi U. Saputra, Andrew Markham, and Niki Trigoni. Visual slam and structure from motion in dynamic environments: A survey. *ACM Computing Surveys*, 51(2), Feb. 2018.

[79] Christoph Schöller, Vincent Aravantinos, Florian Lay, and Alois Knoll. What the constant velocity model can teach us about pedestrian motion prediction. *IEEE Robotics and Automation Letters*, 5(2):1696–1703, 2020.

[80] Script Hook V. `http://www.dev-c.com/gtav/`.

[81] Sarthak Sharma, Junaid Ahmed Ansari, J Krishna Murthy, and K Madhava Krishna. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In *Proc. ICRA*, pages 3508–3515. IEEE, 2018.

[82] Ramana Sundararaman, Cedric De Almeida Braga, Eric Marchand, and Julien Pettre. Tracking pedestrian heads in dense crowd. In *Proc. CVPR*, pages 3865–3875, 2021.

[83] Lei Tai, Jingwei Zhang, Ming Liu, and Wolfram Burgard. Socially compliant navigation through raw depth inputs with generative adversarial imitation learning. In *Proc. ICRA*, pages 1111–1117. IEEE, 2018.

[84] Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Robotics research*, pages 3–19. Springer, 2011.

[85] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. NeurIPS*, pages 5998–6008, 2017.

[86] Jonathan Vincent, Mathieu Labbé, Jean-Samuel Lauzon, François Grondin, Pier-Marc Comtois-Rivet, and François Michaud. Dynamic object tracking and masking for visual slam. In *Proc. IROS*, pages 4974–4979. IEEE, 2020.

[87] Peter M. Visscher. Sizing up human height variation. *Nature Genetics*, 40:489–490, 2008.

[88] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. In *Proc. CVPR*, pages 8198–8207, 2019.

[89] Tingwu Wang, Xuchan Bao, Ignasi Clavera, Jerrick Hoang, Yeming Wen, Eric Langlois, Shunshi Zhang, Guodong Zhang, Pieter Abbeel, and Jimmy Ba. Benchmarking model-based reinforcement learning. *arXiv preprint arXiv:1907.02057*, 2019.

[90] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. In *Proc. ECCV*, 2020.

[91] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *Proc. IROS*, pages 10359–10366. IEEE, 2020.

[92] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *Proc. ICCV*, pages 10033–10041, 2021.

[93] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. Pose Flow: Efficient online pose tracking. In *Proc. BMVC*, 2018.

[94] Takuma Yagi, Karttikeya Mangalam, Ryo Yonetani, and Yoichi Sato. Future person localization in first-person videos. In *Proc. CVPR*, pages 7593–7602, 2018.

[95] Shichao Yang and Sebastian Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, 35(4):925–938, 2019.

[96] Weixiang Yang, Qi Li, Wenxi Liu, Yuanlong Yu, Yuexin Ma, Shengfeng He, and Jia Pan. Projecting your view attentively: Monocular road scene layout estimation via cross-view transformation. In *Proc. CVPR*, pages 15536–15545, 2021.

[97] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Qiao Fei. Ds-slam: A semantic visual slam towards dynamic environments. In *Proc. IROS*, pages 1168–1174. IEEE, 2018.

[98] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proc. ICCV*, 2021.

[99] Jianfeng Zhang, Dongdong Yu, Jun Hao Liew, Xuecheng Nie, and Jiashi Feng. Body meshes as points. *Proc. CVPR*, 2021.

[100] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *Proc. CVPR*, pages 13760–13769, 2022.